

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원번호 : 10-2003-0011121
Application Number

출원년월일 : 2003년 02월 21일
Date of Application
FEB 21, 2003

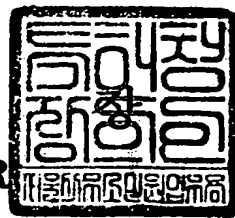
출원인 : 주식회사 하이닉스반도체
Applicant(s) Hynix Semiconductor Inc.



2003 년 07 월 07 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0003
【제출일자】	2003.02.21
【발명의 명칭】	고속 데이터엑세스를 위한 반도체 메모리장치 및 그 구동 방법
【발명의 영문명칭】	SEMICONDUCTOR MEMORY DEVICE FOR HIGH SPEED DATA ACCESS AND METHOD FOR OPERATING THE SAME
【출원인】	
【명칭】	주식회사 하이닉스반도체
【출원인코드】	1-1998-004569-8
【대리인】	
【명칭】	특허법인 신성
【대리인코드】	9-2000-100004-8
【지정된변리사】	변리사 정지원, 변리사 원석희, 변리사 박해천
【포괄위임등록번호】	2000-049307-2
【발명자】	
【성명의 국문표기】	안진홍
【성명의 영문표기】	AHN, Jin Hong
【주민등록번호】	581124-1110419
【우편번호】	431-070
【주소】	경기도 안양시 동안구 평촌동 130-1 영풍아파트 101-1408
【국적】	KR
【발명자】	
【성명의 국문표기】	홍상훈
【성명의 영문표기】	HONG, Sang Hoon
【주민등록번호】	700930-1064113
【우편번호】	467-860
【주소】	경기도 이천시 부발읍 신하리 청구아파트 101-1302
【국적】	KR

【발명자】

【성명의 국문표기】 김세준
【성명의 영문표기】 KIM, Se Jun
【주민등록번호】 740209-1069419
【우편번호】 463-500
【주소】 경기도 성남시 분당구 구미동 까치마을 선경아파트
 107-1002
【국적】 KR

【발명자】

【성명의 국문표기】 고재범
【성명의 영문표기】 KO, Jae Bum
【주민등록번호】 760926-1026025
【우편번호】 131-208
【주소】 서울특별시 중랑구 면목8동 2-8 3층 6/6
【국적】 KR

【우선권주장】

【출원국명】 KR
【출원종류】 특허
【출원번호】 10-2002-0066269
【출원일자】 2002.10.29
【증명서류】 첨부

【심사청구】

청구

【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인
 특허법인 신성 (인)

【수수료】

【기본출원료】 20 면 29,000 원
【가산출원료】 137 면 137,000 원
【우선권주장료】 1 건 26,000 원
【심사청구료】 79 항 2,637,000 원
【합계】 2,829,000 원

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 고속으로 데이터를 액세스할 수 있는 메모리 장치 및 그의 구동방법을 제공하기 위한 것으로, 이를 위해 본 발명은 다수개의 단위셀을 구비하는 제1 셀블럭; 다수개의 단위셀을 구비하는 제2 셀블럭; 및 상기 제1 셀블럭에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 상기 제1 셀블럭에서는 상기 제1 데이터의 재저장 동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 제2 셀블럭에서는 상기 제1 데이터의 재저장 동작이 수행되도록 제어하기 위한 제어수단을 구비하는 메모리 장치를 제공한다.

【대표도】

도 5

【색인어】

반도체, 메모리, 센스앰프, 예비워드라인, 재저장, 단위셀블럭, 셀블럭어드레스

【명세서】**【발명의 명칭】**

고속 데이터억세스를 위한 반도체 메모리장치 및 그 구동방법{SEMICONDUCTOR MEMORY DEVICE FOR HIGH SPEED DATA ACCESS AND METHOD FOR OPERATING THE SAME}

【도면의 간단한 설명】

도1은 종래기술에 의한 메모리 장치의 대략적인 블럭구성도이다.

도2는 도1에 도시된 비트라인 센스앰프부 및 셀블럭의 일예를 나타내는 회로도이다.

도3은 도1에 도시된 메모리 장치에서 한 뱅크에 연속해서 데이터를 리드할 때의 동작파형도이다.

도4는 도1에 도시된 메모리 장치에서 뱅크간 인터리빙 동작을 나타내는 파형도이다.

도5는 본 발명에 따른 메모리 장치를 나타내는 블럭구성도이다.

도6는 도5에 도시된 메모리장치에서 제1 셀블럭에 데이터가 연속적으로 액세스되는 경우에 동작타이밍을 나타내는 도면이다.

도7는 도5에 도시된 메모리 장치에서 제1 셀블럭과 제2 셀블럭에 교대로 데이터가 액세스되는 되는 경우의 동작타이밍을 나타내는 도면이다.

도8은 본 발명의 메모리 장치를 구현한 블럭구성도이다.

도9는 도8에 도시된 메모리 장치의 셀영역을 보다 자세히 나타낸 도면이다.

도10은 도9의 도시된 셀블럭의 일부분을 나타내는 회로도이다.

도11a 내지 도11d는 도8에 도시된 로컬비트라인 센스앰프(LSA)와 글로벌비트라인 센스앰프(GSA)간의 데이터 이동을 보여주는 도면이다.

도12는 도8의 메모리 장치에서 하나의 글로벌비트라인 센스앰프부만을 구비하여 고속으로 데이터 액세스가 가능하도록 구성된 본 발명의 메모리 장치의 블럭구성도이다.

도13은 도12에 도시된 셀블럭의 일부분을 나타내는 회로도이다.

도14는 본 발명의 메모리 장치를 나타내는 블럭구성도로서, 입력되는 어드레스에 대응하는 셀블럭과 데이터 재저장을 위한 여분의 셀블럭을 가지는 메모리 장치의 블럭구성도이다.

도15는 도14에 도시된 메모리 장치의 동작을 나타내는 도면이다.

도16은 본 발명에 의해서 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하기 위한 태그블럭과, 예비워드라인에 대한 정보를 저장하고 있는 예비셀블럭 테이블을 구비하는 메모리 장치의 블럭구성도이다.

도17은 도16에 도시된 태그블럭을 나타내는 블럭구성도이다.

도18은 도16에 도시된 예비셀블럭 테이블의 블럭구성도이다.

도19는 도16에 도시된 제어부(420)의 일부를 나타내는 블럭구성도이다.

도20은 도17에 도시된 9개의 단위태그테이블의 일예를 나타내는 회로도이다.

도21는 도18에 도시된 예비셀블럭테이블의 일예를 나타내는 회로도이다.

도22은 도16에 도시된 본 발명에 의한 메모리 장치의 동작타이밍도이다.

도23은 도16에 도시된 본 발명의 메모리 장치에서 태그관련 블록에 대한 동작타이밍도이다.

도24은 본 발명의 바람직한 실시예에 따라 구현한 메모리 장치를 나타내는 블록구성도이다.

도25는 도24의 메모리 장치에서 서로 다른 단위셀블록으로 데이터를 교대로 액세스할 경우, 인터리브 모드에서의 동작을 보여주는 파형도이다.

도26은 도24의 메모리 장치에서 하나의 단위셀블록에서 데이터를 연속적으로 리드할 때의 동작을 보여주는 파형도이며,

도27은 도24의 메모리 장치에서 하나의 단위셀블록에 데이터를 연속적으로 라이트할 때의 동작을 보여주는 파형도이다.

도28은 도25의 메모리 장치에서 도26에 도시된 순서대로 리드명령어가 입력될 때의 시뮬레이션 파형도이다.

도29는 도24에 도시된 글로벌비트라인 연결부를 제어하기 위한 신호를 생성하기 위한 회로도의 일예이다.

도30은 도29에 도시된 회로도에서 생성되는 신호에 따라 본 발명의 메모리 장치가 동작하는 것을 보여주는 파형도이다.

도31은 도23의 메모리 장치에서 강제프리차지 시간(t_{RP})과 셀블럭어드레스를 변환하는 시간(t_{BAT})만큼이 줄어든 로우사이클타임을 가지는 메모리 장치를 나타내는 블록구성도이다.

도32는 도31에 도시된 메모리 장치를 보다 구체적으로 나타낸 블록구성도이다.

도33은 도32에 도시된 메모리 장치가 연속적으로 리드명령어를 수행하는 것을 나타내는 동작 파형도이다.

도34는 도32에 도시된 메모리 장치의 동작을 쉽게 이해하기 위해 도16에 도시된 메모리장치의 동작과 비교하여 나타낸 동작타이밍도이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <35> 본 발명은 반도체 메모리 장치에 관한 것으로, 특히 데이터를 고속으로 액세스(Access)할 수 있는 메모리 장치에 관한 것이다.
- <36> 반도체 메모리 장치는 크게 RAM(Random Access Memory)과 ROM(Read only Memory)으로 구분할 수가 있다.
- <37> 램(RAM)은 1개의 트랜지스터(transister)와 1개의 캐패시터가 하나의 단위셀(unit cell)을 구성하는 다이내믹 램(Dynimic RAM)과, 6개의 트랜지스터 또는 4개의 트랜지스터 및 2개의 부하 저항으로 구성되는 스태틱램(Static RAM)으로 나뉘어 지는데, 집적도 면에서나 제조공정등에서 효율적인 다이내믹 램이 컴퓨터의 메인 메모리등 여러분야에 널리 사용되고 있다.
- <38> 근래에 중앙처리장치(CPU)의 동작속도는 메모리 장치(DRAM)의 동작속도를 능가할 정도로 현저히 향상되어 왔으며, 그 결과 메모리 장치의 동작속도가 중앙처리장치의 동작속도보다 상대적으로 느려 여러가지 문제점이 발생하고 있다. 이러한 문제점을 극복

하기 위해 보다 고속으로 데이터를 입출력하기 위한 다양한 구조의 메모리 장치가 개발되고 있다.

<39> 도1은 종래기술에 의한 메모리 장치의 대략적인 블럭구성도이다.

<40> 도1을 참조하여 살펴보면, 메모리 장치는 외부에서 입력되는 다수의 명령어신호(/RAS,/CAS,/WE,/CS,CKE,CK등)를 입력받아 뱅크(100)에서 수행되는 동작(예컨대 리드(read), 라이트(write), 리프레쉬(refresh) 동작)을 제어하기 위한 명령어제어부(200)와, 독립적으로 로우디코더(row decoder)와 칼럼디코더(column decoder)를 구비하여, 입력되는 어드레스에 대응하는 단위셀의 데이터를 리드(read)하여 출력하거나 상기 단위셀에 입력되는 데이터를 라이트(write)하는 동작을 수행하기 위한 뱅크(100)와, 뱅크(100)로부터 입,출력되는 데이터를 버퍼링하여 외부로 입출력하기 위한 데이터 입,출력버퍼(300)를 구비한다.

<41> 통상 메모리 장치는 다수의 뱅크(예컨대 4개의 뱅크)를 구비하고 있는데, 각각의 뱅크는 같은 구조를 가지고 있기 때문에, 도1에는 하나의 뱅크(100)만 도시하였다. 또한 도1은 메모리 장치에 구비되는 블럭중에서 본 발명을 설명하기 위해 필요한 최소한의 블럭만을 도시한 것이다.

<42> 하나의 뱅크(100)에는 각각 다수의 셀블럭을 구비하고 있는 8개의 세그먼트(120a ~ 120h)와, 세그먼트(120a ~ 120d)에서 출력되는 데이터를 증폭하여 데이터 입,출력버퍼(300)로 전달하거나, 데이터입출력버퍼(300)에서 입력되는 데이터를 세그먼트로 전달하기 위한 I/O 센스앰프부(110,130)를 구비하고 있다.

- <43> 하나의 세그먼트(예컨대 120a)는 로우어드레스를 디코딩(decoding)하여 셀영역(120a_1)으로 출력하는 로우어드레스 디코더부(121)와, 컬럼어드레스를 디코딩하여 셀영역(120a_1)으로 출력하는 컬럼어드레스 디코더부(122)와, 다수의 단위셀로 구성되는 셀영역(120a_1)을 구비하고 있다.
- <44> 셀영역(120a_1)은 통상적으로 다수개의 단위셀을 각각 구비하는 다수의 셀블럭(Cell block)(124a~124h)과, 셀블럭에서 출력되는 데이터를 감지 증폭하기 위해 셀블럭(124a~124h)간에 배치된 비트라인 센스앰프부(123a~123i)를 구비하고 있다.
- <45> 도1에 도시된 메모리 장치는 용량이 256Mb인 경우를 나타내고 있는데, 4개의 뱅크로 구성되는 경우 하나의 뱅크에는 64Mb의 단위셀을 구비하며, 하나의 세그먼트는 8Mb로 총 8개의 세그먼트가 하나의 뱅크를 구성한다. 하나의 세그먼트(예컨대 123a)에는 총 8개의 셀블럭(124a~124h)을 구비하고 있고, 하나의 셀블럭(예컨대 124a)은 256개의 워드라인(Word Line)과 4K($4 * 1024$)개의 비트라인을 구비한다. 또한, 하나의 셀블럭에는 $256 * 4K$ 개의 단위셀(예컨대 124a)을 구비한다. 이하에서는 전술한 바와 같이 하나의 세그먼트가 8Mb이고, 8개의 셀블럭이 각각 256개의 워드라인을 구비한 것으로 가정하고 설명한다.
- <46> 또한 8개의 셀블럭(124a~124h) 사이에 9개의 비트라인 센스앰프부(123a~123i)가 구비되어, 하나의 센스앰프부(예컨대 123b)는 이웃한 두 셀블럭(124a, 124b)에 공유되도록 구성되어 있다. 셀블럭의 회로구조상 하나의 셀블럭에는 일측과 타측에 각각 2개의 센스앰프부가 필요하기 때문에 8개의 셀블럭에는 총 16개의 비트라인 센스앰프부가 필요하지만, 회로면적등의 효율을 위해서 하나의 비트라인 센스앰프부를 2개의 이웃한 셀블럭이 공유하고, 셀블럭(예컨대 124a와 124b)과 비트라인 센스앰프(123b)간에 연결부를 두고,

적절한 타이밍에 이웃한 2개의 셀블럭(124a와 124b)중 하나와 비트라인 센스앰프부(123b)를 연결 또는 분리하도록 하고 있다.

<47> 도2는 도1에 도시된 비트라인 센스앰프부 및 셀블럭의 일예를 나타내는 회로도로서, 특히 제1 셀블럭 및 제2 셀블럭(124a,124b)과 비트라인 센스앰프(123b)의 일부분을 도시하고 있다.

<48> 도2를 참조하여 살펴보면, 제1 셀블럭(124a)에는 하나의 모스트랜지스터와 하나의 캐패시터로 구성된 단위셀이 256 * 4k개 구비되어 있고, 워드라인(WL)이 각 단위셀을 구성하는 모스트랜지스터의 게이트단으로 연결되고, 비트라인(BL, /BL)이 워드라인과 교차하면서 단위셀을 구성하는 모스트랜지스터의 드레인단으로 연결되어 있다. 단위셀을 구성하는 각각의 모스트랜지스터는 소스단이 단위셀을 구성하는 캐패시터의 일측과 각각 연결되고, 캐패시터의 타측은 셀플레이트 전원(예컨대 접지전원)과 각각 연결되어 있다. 도시되지 않았지만 제2 셀블럭(124b)에서도 제1 셀블럭(124a)과 같은 구성이다.

<49> 비트라인 센스앰프부(123b)는 비트라인 센스앰프 인에이블신호(RT0, /S)에 의해 인에이블되어, 비트라인(예컨대 BL0, /BL0)에 인가되는 신호 차이를 증폭하기 위한 비트라인 센스앰프(123b_4)와, 비트라인 센스앰프(123b_4)가 디스에이블일 때에 출력되는 프리차지 인에이블신호(BLEQ')에 인에이블되어 비트라인 프리차지 전압(Vblp)으로 비트라인(예컨대 BL0, /BL0)을 프리차지하기 위한

프리차지부(123b_3)와, 이퀄라이제이션 신호(BLEQ)에 의해 인에이블되어 제1 셀블럭(124a)에 연결된 한쌍의 비트라인(예컨대 BL0, /BL0) 전압레벨을 같게 하기 위한 이퀄라이제이션부(123b_2)와, 칼럼어드레스에 의해 생성되는 컬럼제어신호(예컨대 CD0)에 의해서 비트라인 센스앰프(123b_4)에 의해 증폭된 데이터신호를 데이터 라인(DB0, /DB0)으로 출력하기 위한 데이터출력부(123b_5)와, 비트라인 센스앰프(123b_4)를 이웃한 셀블럭(124a, 124b)과 선택적으로 연결하기 위한 연결부(123b_1, 123b_6)를 구비한다.

<50> 여기서 비트라인 센스앰프부(123b)에 구비되는 센스앰프의 수는 이웃한 셀블럭에 구비되는 비트라인쌍 수에 따라 정해지고, 센스앰프 연결신호(BISH, BISL)에 의해서 제1 셀블럭(124a) 또는 제2 셀블럭(124b)에 비트라인 센스앰프부(123b)가 연결된다. 도2의 비트라인쌍(BL1, /BL1)은 도시되지는 않았지만 비트라인 센스앰프부(도1의 123a 참조)에 구비된 센스앰프에 연결된다.

<51> 도3 및 도4는 도1에 도시된 메모리 장치의 동작을 나타내는 파형도이다.

<52> 이하에서는 도1 내지 도4를 참조하여, 종래 기술에 의한 메모리 장치의 동작을 살펴본다.

<53> 먼저, 메모리 장치에 입력되는 어드레스를 살펴보면, 어드레스는 크게 뱅크어드레스, 로우어드레스, 컬럼어드레스로 구분되고, 로우어드레스는 셀블럭어드레스와 로컬어드레스로 구분된다. 뱅크어드레스는 4개의 뱅크중 하나를 선택하기 위한 어드레스이며, 셀블럭어드레스는 하나의 세그먼트와 선택된 세그먼트내의 셀블럭을 선택하기 위한 어드레스 신호이다. 로컬어드레스는 셀블럭내에서 하나의 워드라인을 선택하기 위한 어드레스 신호이며, 컬럼어드레스는 하나의 워드라인에 의해 선택된 4K개의 단위셀중 하나를 선택하기 위한 신호이다.

- <54> 계속해서 메모리 장치의 리드동작에 대해서 살펴본다.
- <55> 리드명령어(예컨대 RD0)와 그에 대응하여 입력된 뱅크어드레스에 의해서 하나의 뱅크(예컨대 100)가 선택되고, 이어서 상기 리드명령어(RD0)에 대응하여 입력된 로우어드레스가 선택된 뱅크의 각 세그먼트에 구비된 로우어드레스 디코더부(예컨대 121)로 입력된다.
- <56> 이어서, 로우어드레스 디코더부(121)에서 입력된 로우어드레스에서 셀블럭어드레스와 로컬어드레스를 감지하고, 감지된 셀블럭어드레스에 의해 뱅크(100)의 한 세그먼트(예컨대 120a)와 세그먼트(120a)내에서 하나의 셀블럭(예컨대 제1 셀블럭(124a))을 선택한다.
- <57> 이어서, 제1 셀블럭에 구비된 비트라인쌍(BL, /BL)이 프리차지부(도2의 123b_3 참조)에 의해 프리차지(통상적으로 전원전압의 1/2)되어 있는 상태에서, 로우어드레스 디코더부(121)에서 출력되는 로컬어드레스에 의해서 256개의 워드라인중 하나의 워드라인(예컨대 제1 셀블럭의 WL0)을 활성화시킨다.
- <58> 이어서 활성화된 워드라인(WL0)에 연결된 4K개의 단위셀에 각각 저장되어 있던 4K개의 데이터가 각각의 비트 라인(BL0, BL1, ..., BL1023*4)에 인가된다. 이 때 도2에 도시된 센스앰프 연결신호(BISH)는 인에이블 상태이고, 센스앰프 연결신호(BISL)는 디스에이블 상태가 되어 비트라인센스앰프부(123b)는 제1 셀블럭(124a)과 연결된 상태이다.
- <59> 이어서 비트라인 센스앰프(123b_4)는 비트라인에 인가된 데이터신호를 감지, 증폭한다.

<60> 여기서 제1 셀블럭(124a)의 일측에 구비된 비트라인 센스앰프부(123b)는 짝수번째의 비트라인(BL0,/BL0,BL2,/BL2,...,BL1022*4,/BL1022*4)에 인가된 데이터신호를 감지 증폭하고, 홀수번째의 비트라인(BL1,/BL1,BL3,/BL3,...,BL1023*4,/BL1023*4)에 인가된 데이터신호는 타측에 구비된 비트라인 센스앰프부(123a)에 의해서 감지, 증폭된다. 이렇게 하나의 비트라인 센스앰프부가 셀블럭사이에 구비되어 이웃한 셀블럭과 적절한 타이밍에 연결또는 분리하도록 하는 이유는 전술한 바와 같이 메모리 장치의 고집적을 위한 것이다.

<61> 이어서, 컬럼어드레스 디코더부(122)는 상기 리드명령어(RD0)에 대응하여 입력된 컬럼어드레스를 디코딩하여 비트라인 센스앰프부(123b)의 컬럼선택신호(CD0,CD1,...)(도2의 123b_5 참조)를 출력하고, 비트라인센스앰프(123b_4)에 의해 증폭이 끝난 4K의 데이터중에서 컬럼선택신호에 의해 선택된 데이터가 데이터라인(DB,/DB)를 통해 I/O 센스앰프부(도1의 110)로 출력된다.

<62> 이어서 I/O 센스앰프(110)는 비교적 긴 데이터 라인으로 인해 줄어들 데이터신호를 한번 더 증폭하여 데이터 입,출력버퍼(300)로 전달하고, 데이터 입,출력버퍼(300)에서는 각 뱅크에서 출력되는 데이터를 프리패치하여 데이터가 출력될 타이밍에 외부로 출력한다.

<63> 한편, 비트라인 센스앰프(123b_4)에 의해 감지, 증폭되어 래치된 4K개의 데이터는 리드명령어(RD0)에 의해 활성화되었던 워드라인(WL0)에 대응되는 4K개의 단위셀에 재저장하는 동작이 이루어진다.

<64> 메모리 장치의 집적도를 위해서 하나의 단위셀을 구성하는 캐패시터는 캐패시턴스를 최대한 작게 수 펨토(f)정도로 제조되고, 이로 인해 하나의 단위셀을 구성하는 캐패

시터의 데이터 신호로 저장되는 전하의 양은 매우 작다. 따라서 단위셀의 캐패시터에 저장된 전하가 비트라인에 인가되고, 센스앰프에 의해 증폭된 후에는 다시 증폭된 신호를 가지고 캐패시터에 재저장하는 동작이 반드시 필요한 것이다. 통상적으로 재저장 동작으로 단위셀을 구성하는 캐패시터 충전용량의 90% 이상이 충전될 수 있도록 하고 있다.

<65> 한편, 메모리 장치의 기억소자로 캐패시터를 사용하기 때문에 주기적으로 재충전해주는 리프레쉬 동작이 필요한데, 상기의 재저장 동작을 오래하면 할수록 리프레쉬 동작의 간격이 길어질 수 있다. 따라서 리프레쉬 동작의 주기를 생각해서는 충분히 재저장동작을 수행해야 한다. 그러나 한번의 명령어에 의해 워드라인을 활성화시키고 나서, 다음 명령어를 입력받아 다른 워드라인을 활성화시키는 시간인 로우사이클 타임 측면에서는 상기의 재저장동작은 작을수록 고속으로 데이터 처리가 가능하기 때문에 재저장동작을 충분히 오랫동안 수행할 수 없게 된다.

<66> 따라서 점점 더 메모리 장치의 동작이 고속화되기 위해서는 데이터 재저장시간을 점점 더 줄여야 하는데, 데이터 재저장시간을 줄이게 되면 리프레쉬 동작간격이 줄어드는 등의 다른 문제점이 생기게 되는 것이다.

<67> 도3은 도1에 도시된 메모리 장치의 리드 동작에 관한 파형도이다.

<68> 도3을 참조하여 살펴보면, 제1 리드명령어(RD0)가 메모리 장치에 입력되면, 첫번째 타이밍(t0)에서는 입력된 로우어드레스를 셀블럭어드레스 및 로컬어드레스로 감지하고, 감지된 셀블럭어드레스 및 로컬어드레스에 대응되는 하나의 셀블럭 및 워드라인(제1 셀블럭의 워드라인(WL0))이 선택되고, 선택된 워드라인(WL0)이 활성화된다. 이어서 활성화된 워드라인(WL0)에 대응하는 4K개의 단위셀에 저장된 데이터가 제1 셀블럭에 연결된 비트라인 센스앰프부(123a, 123b)에 의해 감지, 증폭된다.

- <69> 어어서 두번째 타이밍(t_1)에서는 입력된 컬럼어드레스에 의해 디코딩된 컬럼선택신호(예컨대 CD_0)에 의해 4K개의 데이터중에서 선택된 데이터(D_0)를 외부로 출력시키고, 다른 한편으로는 활성화된 워드라인(WL_0)에 대응하는 4K개의 데이터에 대한 재저장 동작을 수행한다. 따라서 하나의 리드명령어를 수행하는데 두번의 타이밍(t_0, t_1)이 필요하다.
- <70> 상기의 두구간(t_0, t_1)이 끝난후에 다시 제2 리드명령어(RD_1)가 입력되면 전술한 바와 같이 두 타이밍(t_2, t_3)동안 한번의 리드동작(RD_1)을 수행한다. 도3에 도시된 노멀로우사이클 타임은 로우어드레스를 입력받아 하나의 워드라인이 활성화되고, 다음 로우어드레스에 의한 워드라인이 활성화될 때까지의 시간을 말한다.
- <71> 이어서 데이터를 라이트하는 동작에 대해 간단하게 살펴본다.
- <72> 데이터를 저장하는 동작은 데이터를 리드하는 동작에서 설명한 워드라인이 활성화되어 4K개의 데이터가 비트라인 센스앰프에 의해 감지 증폭되는 과정은 똑같다.
- <73> 첫번째 타이밍에서 로우어드레스에 의해 하나의 워드라인이 활성화되고, 활성화된 하나의 워드라인에 대응하는 4K개의 데이터가 비트라인 센스앰프에 의해 감지, 증폭한다.
- <74> 이어서 두번째 타이밍에서 컬럼어드레스에 의해 디코딩된 컬럼선택신호에 의해 해당되는 비트라인 센스앰프에 외부에서 입력된 데이터가 교체 저장되고, 이후에 활성화된 워드라인에 해당하는 4K개의 데이터를 재저장하는 동작이 수행된다.

- <75> 이 때에 센스앰프에 교체저장된 데이터는 외부에서 입력되는 큰 신호이기 때문에 다시 감지 증폭할 필요는 없다. 따라서 데이터를 라이트하는 동작도 도3에 도시된 두 타이밍(t_0, t_1)에 의해 진행되는 것이다.
- <76> 결국, 캐패시터를 저장매체로 하는 메모리 장치는 그 특성상 데이터를 리드하던지 또는 라이트하던지, 첫번째로는 단위셀에 있는 데이터를 감지, 증폭하는 타이밍과, 두번째로는 증폭된 데이터를 외부로 출력하거나 외부에서 입력된 데이터로 교체한 다음, 데이터가 있던 단위셀에 재저장하는 타이밍이 필요한 것이다.
- <77> 고집적 메모리 장치에서 데이터 저장 매체인 캐패시터의 크기가 매우 작아서 데이터신호에 따라 저장되는 전하의 수가 매우 적게 되고, 이로 인해 캐패시터에 저장된 전하를 감지, 증폭하게 되면 캐패시터에서는 저장된 데이터신호가 파괴되는데 전술한 바와 같이 데이터 재저장동작은 한번의 리드 또는 라이트 명령어 동작후에 데이터를 다시 보존하기 위해 하는 동작이다.
- <78> 따라서 하나의 명령어에 따라 데이터를 액세스하고 난 후, 바로 다음 명령어를 입력받아 다음 데이터를 액세스할 수 있는 것이 아니라 이전 데이터에 대한 재저장동작이 끝난 후에야 다음 데이터에 대한 액세스가 가능한 것이다. 즉, 데이터를 계속 유지하기 위한 재저장 시간으로 인해 데이터를 액세스하는 데 필요한 실질적인 시간보다 더 많은 시간이 소요되는 것이다.
- <79> 실제로 캐피시터를 데이터 저장 매체로 사용하는 메모리 장치에서는 데이터를 액세스하는 시간은 어드레스를 입력받아 하나의 워드라인을 선택하여 활성화하거나, 감지 증폭된 데이터를 외부로 출력하는 시간 보다는 수펄토 정도의 캐패시터에서 데이터를 감지, 증폭하는 시간과 데이터를 다시 재저장하는 시간이 대부분을 차지하게 된다.

- <80> 그러므로 캐패시터를 데이터 기억매체로 사용하는 메모리 장치를 보다 고속으로 동작하기 위해서는 데이터 재저장 시간이 데이터 액세스시간에 영향을 주지 않는 구조 및 방법을 개발하는 것이 필요하다.
- <81> 한편, 보다 고속으로 데이터를 액세스하기 위해 데이터 재재장시간만큼 데이터 액세스 시간을 줄일 수 있는뱅크간 인터리브 모드(Interleave Mode)를 사용하는 메모리 장치가 제안되었다.
- <82> 뱅크간 인터리브 모드를 사용하는 메모리 장치란 일정시간 안에 많은 데이터를 전송하기 위한 뱅크 인터리빙(Bank Interleaving) 방법을 사용되는데, 한 뱅크에서 데이터를 출력하고 재저장하는 동안에 이웃한 뱅크에서 데이터가 출력되어, 외부에서 보면 재저장하는 시간없이 연속적으로 명령어를 입력받아 데이터가 연속적으로 출력되는 것으로 보이게 하는 것이다.
- <83> 도4에는 뱅크간 인터리빙 동작을 나타내는 파형도이다.
- <84> 도4를 참조하여 살펴보면, 인터리빙 동작시에 메모리 장치는 첫번째 리드명령어(RD0)에 의해 뱅크0의 한 워드라인(B0의 WL0)이 활성화되고, 다음 구간에 데이터를 출력시키고 재저장하는 한편, 다른 뱅크1에서 리드명령어(RD1)를 연속적으로 입력받아 한 워드라인(B1의 WL0)을 활성화시키고 데이터(D1)를 출력시킨다. 따라서 인터리빙 모드로 동작하게 되면 연속적으로 리드명령어(RD0, RD1, RD2)를 입력받아 데이터(D0, D1, D2)를 연속적으로 출력할 수 있는 것이다.

- <85> 즉 한 बैं크에서 데이터를 재저장하는 동안 이웃한 बैं크에서는 명령어를 입력받아 데이터를 감지, 증폭하여 출력할 수 있기 때문에 데이터 재저장하는 시간만큼 데이터 액세스시간을 줄일 수 있는 것이다. बैं크간 인터리빙을 동작시키면 도4에 도시된 바와 같이 로우 사이클 타임을 노멀 로우 사이클 타임보다 크게 줄일 수 있다.
- <86> 그러나, 인터리빙 모드를 사용하는 메모리 장치에서도 동일 बैं크에 연속적으로 액세스가 집중되는 경우에는 인터리빙동작을 수행할 수 없게 되어 고속으로 데이터를 입출력시킬 수 없는 문제점을 가지고 있다.
- <87> 입력되는 명령어에 의해 한 बैं크만 연속적으로 데이터를 액세스할 때에는 बैं크간 인터리빙 모드로 동작할 수 없게 되고, 전술한 바와 같이 이전 명령어에 의한 데이터 재저장시간이 끝난후에 다시 명령어를 입력받을 밖에 없는 것이다.
- <88> 따라서 बैं크간 인터리빙을 사용하는 메모리 장치라도 입력되는 명령어가 한 बैं크의 데이터만을 연속적으로 액세스하는 패턴인지, 또는 बैं크를 옮겨가면서 데이터를 액세스하는 패턴인지에 따라 그 동작속도가 크게 달라지는 것이다.
- <89> 액세스되는 데이터의 패턴에 따라 메모리 장치의 액세스 속도가 크게 영향을 받게 되는 메모리 장치를 시스템에 사용하게 되면 시스템의 동작속도에 차이가 생겨 시스템의 동작신뢰도에 문제점을 발생시킬 수 있다.
- <90> 따라서 데이터 액세스 패턴에 관계없이 데이터를 고속으로 액세스할 수 있는 메모리 장치가 필요하다.

【발명이 이루고자 하는 기술적 과제】

- <91> 본 발명은 상기의 문제점을 해결하기 위해 제안된 것으로, 데이터 재저장시간이 데이터 액세스시간에 영향을 주지 않아, 고속으로 데이터를 액세스할 수 있는 메모리 장치 및 그의 구동방법을 제공하는 것을 목적으로 한다.
- <92> 또한 본 발명은 인터리빙모드로 동작하면서, 한 뱅크내에서 연속적인 데이터 액세스가 발생하는 경우에 인터리빙모드에 비해 데이터 액세스 속도의 저하없이 고속으로 동작함으로써, 데이터 액세스 패턴에 관계없이 고속으로 데이터를 액세스할 수 있는 메모리 장치 및 그의 구동방법을 제공하는 것을 목적으로 한다.

【발명의 구성 및 작용】

- <93> 상기의 목적을 달성하기 위한 본 발명은 다수개의 단위셀을 구비하는 제1 셀블럭; 다수개의 단위셀을 구비하는 제2 셀블럭; 및 상기 제1 셀블럭에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 상기 제1 셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 제2 셀블럭에서는 상기 제1 데이터의 재저장 동작이 수행되도록 제어하기 위한 제어수단을 구비하는 메모리 장치를 제공한다.
- <94> 또한, 본 발명은 다수의 단위셀을 구비하는 단위셀블럭과, 상기 단위셀블럭의 일측에 구비되어 상기 단위셀블럭의 일부 데이터를 감지하여 증폭하기 위한 제1 로컬비트라인 센스앰프부와, 상기 단위셀블럭의 타측에 구비되어 상기 단위셀블럭의 나머지 데이터를 감지하여 증폭하기 위한 제2 로컬비트라인 센스앰프를 구비하는 제1 기본셀블럭; 상

기 제1 기본셀블럭과 같은 구성을 가지는 제2 내지 제4 기본셀블럭; 상기 제1 내지 제4 기본셀블럭에 각각 구비된 제1 로컬비트라인 센스앰프부에서 감지 증폭된 데이터를 래치하기 위한 제1 글로벌비트라인 센스앰프부; 상기 제1 내지 제4 기본셀블럭에 각각 구비된 제2 로컬비트라인 센스앰프부에서 감지 증폭된 데이터를 래치하기 위한 제2 글로벌비트라인 센스앰프부; 상기 제1 기본셀블럭의 제2 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부로 연결하거나, 상기 제2 기본셀블럭의 제1 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부와 선택적으로 연결하기 위한 제1 글로벌비트라인 연결부; 상기 제3 기본셀블럭의 제2 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부로 연결하거나, 상기 제4 기본셀블럭의 제1 로컬비트라인 센스앰프부를 상기 제1 글로벌비트라인 센스앰프부와 선택적으로 연결하기 위한 제1 글로벌비트라인 연결부; 및 제1 글로벌비트라인 센스앰프부와 상기 제2 글로벌비트라인 센스앰프부에 래치된 데이터를 재저장 동작에 사용되도록 제어하기 위한 제어수단을 구비하는 메모리 장치를 제공한다.

<95> 또한 본 발명은 다수의 단위셀을 각각 구비하고, 상기 다수의 단위셀에 저장된 데이터를 감지증폭하기 위한 로컬비트라인 센스앰프부를 각각 구비하는 다수의 기본셀블럭; 상기 기본셀블럭에 각각 구비된 다수의 로컬비트라인 센스앰프부에 의해 감지 증폭된 데이터를 래치하기 위한 글로벌비트라인 센스앰프부; 및 상기 래치된 데이터를 재저장 동작에 사용되도록 제어하기 위한 제어수단을 구비하는 메모리 장치를 제공한다.

<96> 또한 본 발명은 각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 N개로 구성된 단위셀블럭에, 추가적으로 M개의 워드라인을 가지는 단위셀블럭을 더

포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭; 및 상기 N+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭으로부터 액세스되는 데이터를 상기 제1 단위셀블럭 또는 상기 제2 단위셀블럭으로 재저장시키도록 제어하는 제어수단을 구비하는 메모리 장치를 제공한다.

<97> 또한 본 발명은 각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 M개의 워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성되는 셀블럭; 상기 (N+1) 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블; 상기 로우어드레스를 입력받아 N개의 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 상기 N+1개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하기 위한 태그블럭; 및 상기 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 상기 워드라인에 대한 -상기 예비셀블럭 테이블로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 상기 태그블럭과 상기 예비셀블럭 테이블을 제어하는 제어수단을 구비하는 메모리 장치를 제공한다.

<98> 또한 본 발명은 각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 M개의 워드라인을 더 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭; 상기 N+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭으로부터 액세스되는 데이터가 상기 제1 단위셀블럭 또는 제2 단위셀블럭으로 재저장되도록 제어하는 데이터엑세스 제어수단; 및 현재 실행되는 제1 명령어에 대응하는 데이터 엑세스 중에, 다음 실행될 제2 명령어에 대응하는 셀블럭어드레스 변환동작과 상기 제1 명령어에 대응하는 비트라인에 대한 강제프리차지동

작이 이루어지도록 상기 데이터엑세스 제어수단을 제어하는 명령어 제어수단을 구비하는 메모리 장치를 제공한다.

<99> 또한 본 발명은 다수개의 단위셀을 각각 구비하는 제1 및 제2 셀블록을 구비하는 메모리 장치의 구동방법에 있어서, 상기 제1 셀블록에서 제1 데이터를 감지, 증폭하는 제1 단계; 상기 제2 셀블록에서 상기 제1 데이터를 재저장하는 제2 단계; 및 상기 제1 셀블록에서 제2 데이터를 감지, 증폭하는 제3 단계를 포함하며, 상기 제2 단계 및 상기 제3 단계는 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법을 제공한다.

<100> 또한 본 발명은 각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블록에, 추가적으로 단위셀블록을 더 포함하여 N+1개의 단위셀블록으로 구성된 셀블록을 구비하는 메모리 장치의 구동방법에 있어서, 상기 N+1개의 단위셀블록중에서 선택된 제1 단위셀블록에서 제1 데이터를 감지, 증폭하는 제1 단계; 상기 N+1개의 단위셀블록중에서 선택된 제2 단위셀블록으로 상기 제1 데이터를 재저장하는 제2 단계; 및 상기 제1 셀블록에서 제2 데이터를 감지, 증폭하는 제3 단계를 포함하며, 상기 제2 단계 및 상기 제3 단계는 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법을 제공한다.

<101> 또한 본 발명은 각각 M개의 워드라인을 가지고, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블록에, 추가적으로 단위셀블록을 더 포함하여 N+1개의 단위셀블록으로 구성된 셀블록을 구비하는 메모리 장치의 구동방법에 있어서, 상기 N+1개의 단위셀블록 중에서 선택된 제1 단위셀블록의 제1 워드라인을 활성화시키는 제1 단계; 상기 제1 워드라인에 대응되는 K개의 데이터를 감지 증폭하는 제2 단계; 감지증폭된 상기 제1

워드라인에 대응되는 K개의 데이터를 상기 제1 워드라인에 대응하는 예비워드라인이 구비된 셀블럭으로 이동시켜 재저장하는 제3 단계; 상기 제1 단위셀블럭에 제2 워드라인을 활성화시키는 제4 단계; 및 상기 제2 워드라인에 대응하는 K개의 데이터를 감지증폭하는 제5 단계를 포함하며, 제3 단계 내지 제5 단계의 수행은 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법을 제공한다.

<102> 또한 본 발명은 각각 M개의 워드라인을 가지고, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 상기 M개의 예비워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭을 구비하는 메모리 장치의 구동방법에 있어서, 상기 로우어드레스를 입력받아 N개의 논리적 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스와, 선택된 단위셀블럭에 구비된 M개의 워드라인중 하나를 선택하기 위한 로컬어드레스로 감지하는 제1 단계; 상기 논리적 셀블럭어드레스를 N+1개의 물리적 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하는 제2 단계; 상기 변환된 물리적 셀블럭어드레스에 대응하여 선택된 제1 단위셀블럭에서 상기 로컬어드레스에 대응하는 제1 워드라인을 활성화시키는 제3 단계; 상기 제1 워드라인에 대응되는 제1 데이터를 감지증폭하는 제4 단계; 상기 제1 데이터를 상기 제1 워드라인에 대응되는 예비워드라인이 지정된 제2 단위셀블럭으로 이동시켜 재저장하는 제5 단계; 다음명령을 위해 입력된 로컬어드레스에 대응하는 상기 제1 단위셀블럭의 제2 워드라인을 활성화시키는 제6 단계; 및 상기 제2 워드라인에 대응하는 제2 데이터를 감지증폭하는 제7 단계를 포함하며, 상기 제5 단계 내지 상기 제7 단계는 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법을 제공한다.

<103> 또한 본 발명은 각각 M개의 워드라인을 가지고, 입력되는 논리적 셀블럭어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 상기 M개의 예비워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭을 구비하는 메모리 장치의 구동방법에 있어서, 제1 명령어에 대응하는 제1 논리적 셀블럭어드레스를 입력받아 상기 N+1개의 단위셀블럭중 하나를 선택하기 위한 제1 물리적 셀블럭어드레스로 변환하는 제1 단계; 상기 제1 물리적 셀블럭어드레스에 대응하는 제1 단위셀블럭에서 제1 데이터를 감지증폭하는 제2 단계; 상기 제1 데이터를 이동시켜 래치하는 제3 단계; 상기 제1 단위셀블럭에서 감지증폭된 상기 제1 데이터를 강제프리차지시키는 제4 단계; 제2 명령어에 대응하는 제2 논리적 셀블럭어드레스를 입력받아 제2 물리적 셀블럭어드레스로 변환하는 제5 단계; 상기 래치된 제1 데이터를 상기 N+1개의 단위셀블럭중에서 선택된 제2 단위셀블럭으로 이동시켜 재저장하는 제6 단계; 및 상기 제2 물리적 셀블럭어드레스에 대응하는 제1 단위셀블럭에서 제2 데이터를 감지증폭하는 제7 단계를 포함하며, 상기 제6 단계 및 상기 제7 단계는 실질적으로 같은 타이밍에 이루어지는 것을 특징으로 하는 메모리 장치의 구동방법을 제공한다.

<104> 이하, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 정도로 상세히 설명하기 위하여, 첨부된 도면을 참조하여 본 발명에 대해서 자세히 설명하기로 한다.

<105> 도5는 본 발명에 따른 메모리 장치를 나타내는 블럭구성도이다.

<106> 도5를 참조하여 살펴보면, 본 발명에 따른 메모리 장치는 다수개의 단위셀을 구비하는 제1 셀블럭(92)과, 다수개의 단위셀을 구비하는 제2 셀블럭(93)과, 제1 셀블럭(92)

에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 제1 셀블럭(92)에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 제2 셀블럭(93)에서 상기 제1 데이터의 재저장 동작이 수행하도록 제어하기 위한 제어부(96)를 구비한다.

<107> 또한, 본 발명의 메모리 장치는 상기 제1 데이터를 제1 셀블럭(60)으로부터 전달받아 래치(latch)하기 위한 래치부(91)와, 래치부(91)에 래치된 데이터를 제2 셀블럭(93)으로 전달시키기 위한 신호라인(94)을 더 구비한다.

<108> 제1 및 제2 셀블럭(92,93)은 상기 제1 데이터 또는 상기 제2 데이터를 감지증폭하기 위한 비트라인 센스앰프를 각각 구비하고, 래치부(91)는 상기 비트라인 센스앰프에서 감지, 증폭되는 데이터를 래치하게 된다.

<109> 또한 제1 및 제2 셀블럭(92,93)은 다수 구비된 단위셀과 상기 비트라인 센스앰프를 각각 연결하기 위한 비트라인을 각각 구비하며, 상기 비트라인 센스앰프와 래치수단(91)간의 데이터 이동은 신호라인(94)을 통하여 이루어진다.

<110> 또한 신호라인(94)이 제1 및 제2 셀블럭(92,93)에 구비된 비트라인 센스앰프와 선택적으로 연결되도록 하는 연결부(95)를 더 구비한다.

<111> 도6는 도5에 도시된 메모리장치에서 제1 셀블럭에 데이터가 연속적으로 액세스되는 경우에 동작타이밍을 나타내는 도면이고, 도7는 제1 셀블럭과 제2 셀블럭에 교대로 데이터가 액세스되는 되는 경우의 동작타이밍을 나타내는 도면이다.

- <112> 이하 도5 내지 7를 참조하여 본 발명의 메모리 장치의 동작을 살펴보는데, 먼저 도6을 참조하여 제1 셀블럭에 연속해서 데이터 액세스가 일어날 때의 메모리 장치의 동작을 살펴본다.
- <113> 도6을 참조하여 살펴보면, 연속적으로 입력되는 명령어(CD0~CD2)('CD0~CD2'은 모두 제1 셀블럭(92)에 있는 데이터를 액세스하는 것으로 가정한다.)에 의해 첫번째 타이밍(t0)에 제1 셀블럭(92)에서 제1 데이터를 감지,증폭하고, 이어서 두번째 타이밍(t1)에 제2 셀블럭(93)에서 상기 제1 데이터를 재저장하고, 다른 한편으로는 제1 셀블럭(92)에서 제2 데이터를 감지, 증폭한다.
- <114> 이어서 세번째 타이밍(t2)에는 제2 셀블럭(93)에서 상기 제2 데이터를 재저장하고, 다른 한편으로는 제1 셀블럭(92)에서 제3 데이터를 감지 증폭한다. 이어서 네번째 타이밍(t3)에서는 제2 셀블럭(93)에서 상기 제3 데이터를 재저장하게 된다.
- <115> 일단 제1 셀블럭에 구비된 비트라인 센스앰프에 의해 데이터가 감지, 증폭되면 래치부(91)로 래치되고, 데이터 재저장시에는 래치부(91)에 래치된 데이터가 제2 셀블럭으로 이동하여 이루어진다. 래치부(91)와 제1 셀블럭(92) 또는 제2 셀블럭(93)간의 데이터 이동은 신호라인(94)을 통해서 이루어지고, 신호라인(94)은 연결부(95)에 의해서 제1 셀블럭(92) 또는 제2 셀블럭(93)과 선택적으로 연결된다.
- <116> 계속에서 도7을 참조하여 제1 셀블럭(92)과 제2 셀블럭(93)에 교대로 데이터 액세스가 일어나는 경우에 대해 살펴본다.
- <117> 연속적으로 입력되는 명령어(CD0~CD2)에 의해 제1 셀블럭(92)과 제2 셀블럭(93)에서 제1 데이터 내지 제3 데이터가 교대로 액세스될 때, 첫번째 타이밍(t0)에 제1

셀블럭(92)에서 제1 데이터를 감지, 증폭하고, 이어서 두번째 타이밍(t1)에 제1 셀블럭(92)에서 상기 제1 데이터를 재저장하고, 다른 한편으로는 제2 셀블럭(93)에서 제2 데이터를 감지, 증폭한다.

<118> 이어서 세번째 타이밍(t2)에는 제2 셀블럭(93)에서 상기 제2 데이터를 재저장하고, 다른 한편으로는 제1 셀블럭(92)에서 제3 데이터를 감지 증폭한다. 이어서네번째 타이밍(t3)에서는 제1 셀블럭(92)에서 상기 제3 데이터를 재저장하게 된다.

<119> 일단 셀블럭에 구비된 비트라인 센스앰프에 의해 감지 증폭된 데이터는 래치부에 래치되고, 데이터 재저장시에는 래치된 데이터가 제1 셀블럭(92) 또는 제2 셀블럭(93)로 이동하여 재저장이 이루어진다. 제1 셀블럭(92)과 제2 셀블럭(93)에 교대로 데이터 액세스가 일어날 셀때에는 블럭간 인터리빙모드로 동작하게 되는데, 감지 증폭되어 래치부(91)에 래치된 데이터가 다시 재저장하는 타이밍에 다음 명령어에 의해 액세스되는 데이터가 감지, 증폭되는 것이다.

<120> 입력된 명령어가 리드명령어일 경우에는 감지 증폭되어 래치부(91)에 래치된 데이터가 외부로 출력되고, 라이트명령어일 경우에는 외부에서 입력된 데이터는 감지 증폭되어 래치된 데이터와 교체된다.

<121> 전술한 바와 같이 메모리장치를 동작시키면, 데이터 재저장하는 시간이 데이터 액세스시간에 영향을 전혀 주지 않아서 고속으로 데이터를 액세스 할 수 있게 된다. 캐패시터를 저장매체로 사용하는 메모리 장치에서는 액세스된 데이터를 감지 증폭하는 시간과 재저장하는 시간이 데이터 액세스시간의 대부분을 차지하므로, 데이터 재저장시간이 데이터 액세스시간에 포함되지 않는다면 데이터 액세스 시간을 크게 줄일수 있게 되는 것이다.

- <122> 또한 본 발명의 메모리 장치는 한 셀블럭에 데이터가 연속적으로 계속 액세스되거나 셀블럭간에 교대로 데이터가 액세스되거나 동일한 로우사이클 타임을 가지게 되어, 액세스되는 데이터 패턴에 상관없이 고속으로 데이터를 액세스할 수 있게 된다.
- <123> 도8은 전술한 바와 같이 감지 증폭된 데이터를 래치시키고, 다음 명령어의 데이터 감지 증폭을 연속적으로 하기 위한 본 발명의 메모리 장치를 나타내는 블럭구성도이다.
- <124> 도8을 참조하여 살펴보면, 본 발명의 메모리 장치는 다수의 단위셀을 구비하는 단위셀블럭(72_1b)과, 단위셀블럭(72_1b)의 일측에 구비되어 단위셀블럭(72_1b)의 일부 데이터를 감지하여 증폭하기 위한 제1 로컬비트라인 센스앰프부(72_1a)와, 단위셀블럭(72_1b)의 타측에 구비되어 단위셀블럭(72_1b)의 나머지 데이터를 감지하여 증폭하기 위한 제2 로컬비트라인 센스앰프(72_1c)를 구비하는 제1 기본셀블럭(72_1)과, 제1 기본셀블럭(72_1)과 같은 구성을 가지는 제2 내지 제4 기본셀블럭(72_2, 72_3, 72_4)과, 제1 내지 제4 기본셀블럭(72_1, 72_2, 72_3, 72_4)에 각각 구비된 제1 로컬비트라인 센스앰프부(72_1a, 72_2a, 72_3a, 72_4a)에서 감지 증폭된 데이터를 래치하기 위한 제1 글로벌비트라인 센스앰프부(71a)와, 제1 내지 제4 기본셀블럭(72_1, 72_2, 72_3, 72_4)에 각각 구비된 제2 로컬비트라인 센스앰프부(72_1c, 72_2c, 72_3c, 72_4c)에서 감지 증폭된 데이터를 래치하기 위한 제2 글로벌비트라인 센스앰프부(71b)와, 제1 기본셀블럭(72_1)의 제2 로컬비트라인 센스앰프부(72_1c)를 제2 글로벌비트라인 센스앰프부(71a)로 연결하거나, 제2 기본셀블럭(72_2)의 제1 로컬비트라인 센스앰프부(72_1a)를 제2 글로벌비트라인 센스앰프부(71b)와 연결하기 위한 제1 글로벌비트라인 연결부(73_1)와, 제3 기본셀블럭(72_3)의 제2 로컬비트라인 센스앰프부(72_3c)를 제2 글로벌비트라인 센스앰프부(71b)로 연결하거나, 제4 기본셀블럭(72_4)의 제1 로컬비트라인 센스앰프부(72_4a)를 제1 글

로컬비트라인 센스앰프부(71a)와 연결하기 위한 제1 글로벌비트라인 연결부(73_2)와, 제1 글로벌비트라인 센스앰프부(71a)와 제2 글로벌비트라인 센스앰프부(71b)에 래치된 데이터를 재저장 동작에 사용되도록 제어하기 위한 제어부(80)를 구비한다.

<125> 또한 본 발명의 메모리 장치는 제1 기본셀블럭(72_1)과 같은 구성을 가지는 상기 제5 기본셀블럭(72_5)와, 제5 기본셀블럭(72_5)의 제2 로컬비트라인 센스앰프부(75_5c)를 제2 글로벌비트라인 센스앰프부(71b)와 연결하기 위한 제3 글로벌비트라인 연결부(73_3)를 더 구비한다.

<126> 또한 본 발명의 메모리 장치는 제1 내지 제5 기본셀블럭(72_1, 72_2, 72_3, 72_4, 72_5)에 각각 구비된 제1 로컬비트라인 센스앰프부(72_1a, 72_2a, 72_3a, 72_4a, 72_5a)에 의해 감지 증폭된 데이터를 제1 글로벌비트라인 센스앰프부(71a)와 연결 -제1 내지 제3 글로벌 비트라인 연결부(73_1, 73_2, 73_3)를 통해서 연결되거나 또는 직접연결- 하거나, 제1 내지 제5 기본셀블럭(72_1, 72_2, 72_3, 72_4, 72_5)에 각각 구비된 제2 로컬비트라인 센스앰프부(72_1c, 72_2c, 72_3c, 72_4c, 72_5c)를 제2 글로벌비트라인 센스앰프부(71b)와 연결 -상기 제1 내지 제3 글로벌 비트라인 연결부(73_1, 73_2, 73_3)를 통해서 연결되거나 또는 직접연결- 하기 위한 글로벌비트라인(74)을 더 구비한다.

<127> 또한 제1 내지 제5 기본셀블럭(72_1, 72_2, 72_3, 72_4, 72_5)에 각각 구비된 제1 및 제2 로컬비트라인 센스앰프(72_1a, 72_2a, 72_3a, 72_4a, 72_5a, 72_1c, 72_2c, 72_3c, 72_4c, 72_5c)부는 제1 내지 제3 글로벌비트라인 연결부(73_1, 73_2, 73_3) 중 하나와, 또는 글로벌비트라인(74)과 연결되기 위한 스위치(도9의 A참조)를 각각 구비한다.

<128> 제어부(80)는 제1 내지 제5 기본셀블럭(72_1, 72_2, 72_3, 72_4, 72_5)중에서 선택된 제1 기본셀블럭(72_1)에 연속적으로 제1 및 제2 데이터가 액세스될 때, 제1 기본셀블

력(72_1)에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 제1 기본셀블럭 이외의 기본셀블럭(예컨대 제2 기본셀블럭(72_2))에서 상기 제1 데이터의 재저장 동작이 수행되도록 제어한다. 참고적으로 도면부호 'BL, /BL'은 기본셀블럭의 로컬비트라인을 나타내는 것이다.

<129> 또한 제어부(80)는 제1 내지 제5 기본셀블럭(72_1, 72_2, 72_3, 72_4, 72_5)중에서 선택된 제1 기본셀블럭(72_1)과 제2 기본셀블럭(72_2)에서 제1 데이터와 제2 데이터가 교대로 액세스 될 때, 상기 제1 데이터를 제1 기본셀블럭(72_1)에 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 제2 기본셀블럭(72_2)에서 상기 제2 데이터가 액세스되도록 제어한다.

<130> 도9는 도8에 도시된 메모리 장치의 셀영역(70)을 보다 자세히 나타낸 도면이다.

<131> 도9를 참조하여 설명하면, 메모리 장치의 셀영역(70)은 다수의 단위셀을 각각 구비하는 제1 내지 제5 기본셀블럭(72_1 ~ 72_5)과, 기본셀블럭 영역의 일측과 타측에 각각 제1 글로벌비트라인 센스앰프부(71a)와 제2 글로벌비트라인 센스앰프부(71b)를 구비하고 있다.

<132> 제1 글로벌비트라인 센스앰프부(71a)와 제2 글로벌비트라인 센스앰프부(71b)는 각각 다수개의 글로벌비트라인 센스앰프(GSA)를 구비하고 있으며, 제1 글로벌비트라인 센스앰프부(71a)에 구비된 하나의 글로벌비트라인 센스앰프(예컨대 71a_1)는 제2 글로벌비트라인 센스앰프부에 구비된 글로벌비트라인 센스앰프(예컨대 71b_1)와 제1 글로벌비트라인(GBL0, /GBL0)으로 연결된다.

- <133> 글로벌비트라인 센스앰프(GSA)는 로컬비트라인 센스앰프(LSA)에서 전달되는 데이터를 래치하는 역할과, 데이터가 글로벌비트라인을 통하여 이동됨으로서 데이터신호가 줄어들게 되는 경우 다시 증폭하는 역할을 하게 된다. 따라서 글로벌비트라인 센스앰프(GSA)는 로컬비트라인 센스앰프(LSA)와 같은 회로구성으로 구현할 수 있다.
- <134> 글로벌비트라인 연결부(73_1 ~ 73_3)는 글로벌비트라인(74)을 이웃한 2개의 로컬비트라인 센스앰프와 선택적으로 연결한다. 글로벌비트라인 연결부(73_1 ~ 73_3)는 2의 기본셀블럭마다 하나씩 구비되는데, 도시된 바와 같이 기본셀블럭이 5개일 경우에는 총 3개가 필요하다. 만약 기본셀블럭이 9개인 경우에는 총 5개의 글로벌비트라인 연결부가 필요하게 된다.
- <135> 하나의 기본셀블럭(예컨대 72_1)은 전술한 바와 같이 각각 제1 및 제2 로컬비트라인 센스앰프(72_1a, 72_1c)와 단위셀블럭(72_1b)으로 구성되는데, 제1 및 제2 로컬비트라인 센스앰프(72_1a, 72_1c)는 각각 다수의 비트라인 센스앰프(LSA)를 구비하고 있다. 로컬비트라인 센스앰프(LSA)는 스위치(도9의 A참조)를 통해 글로벌비트라인(74)과 연결되거나 또는 글로벌비트라인 연결부(73_1)로 연결된다.
- <136> 제1 내지 제5 기본셀블럭 제어부(76a ~ 76e)는 각각의 기본셀블럭에 구비된 로컬비트라인 센스앰프(LSA)가 글로벌비트라인(74) 또는 글로벌비트라인 연결부(73_1, 73_2, 73_3)와 적절한 타이밍에 연결될 수 있도록 스위치 제어신호(예컨대 GBIS12)를 출력하거나 로컬비트라인 센스앰프부에 구비된 스위치를 제어하는 신호를 출력하게 된다.
- <137> 제1 및 제2 글로벌비트라인 센스앰프부(71a, 71b)에 연결된 데이터버스(75a, 75b)는 제1 및 제2 글로벌비트라인 센스앰프부(71a, 71b)에 래치된 데이터를 외부로 출력하거나

외부에서 입력되는 데이터를 제1 및 제2 글로벌비트라인 센스앰프부(71a,71b)에 전달하는 역할을 한다.

- <138> 도10은 도9의 도시된 기본셀블럭의 일부분을 나타내는 회로도로서, 특히 제1 글로벌비트라인 연결부(73_1)의 일부분과, 제1 글로벌비트라인 연결부(73_1)와 이웃한 영역의 제1 및 제2 기본셀블럭(72_1,72_2)을 나타낸 것이다.
- <139> 도10를 참조하여 살펴보면, 제1 및 제2 기본셀블럭(72_1,72_2)에 각각 구비된 단위 셀블럭(72_1b, 72_2b)에는 각각 다수의 워드라인(WL0,WL1,...)과, 로컬비트라인(BL0,/BL0,...)과, 각각의 워드라인과 비트라인에 대응하여 연결되는 다수의 모스트랜지스터와 캐패시터를 구비하고 있다.
- <140> 제1 기본셀블럭(72_1)의 제2 로컬비트라인 센스앰프부(72_1c)는 비트라인센스앰프인에이블신호(RT0, /S)에 의해 인에이블되어 로컬비트라인(예컨대 BL0, /BL0)에 인가된 데이터 신호를 감지, 증폭하기 위한 로컬비트라인 센스앰프(72_1c_3)와, 로컬비트라인 센스앰프(72_1c_3)에 의해 감지, 증폭된 데이터 신호를 글로벌비트라인 연결부(73_1)를 통해 글로벌비트라인(GBL0,/GBL0)으로 전달하기 위한 로컬비트라인 연결부(72_1c_4)를 구비한다.
- <141> 또한, 제2 로컬비트라인 센스앰프부(72_1c)는 로컬비트라인 센스앰프(72_1c_3)가 디스에이블일 때에 출력되는 프리차지인에이블신호(BLEQ)에 인에이블되어 비트라인 프리차지 전압(Vb1p)으로 비트라인(예컨대 BL0, /BL0)을 프리차지하기 위한 프리차지부(72_1c_2)와, 이퀄라이제이션 신호(BLEQ')에 의해 비트라인(BL0,/BL0)의 전압레벨을 같게 하기 위한 이퀄라이제이션부(72_1c_1)를 구비한다.

- <142> 제1 로컬비트라인 센스앰프부(72_1a)는 기본적으로 제2 로컬비트라인 센스앰프부(72_1c)와 같은 구성으로, 로컬비트라인 센스앰프(72_1a_3)와, 로컬비트라인 연결부(72_1a_4)와, 프리차지부(72_1a_2)와, 이퀄라이제이션부(72_1a_1)를 구비한다. 제1 기본셀블럭(72_1)의 비트라인(BL1,/BL1)에 인가된 데이터신호는 기본셀블럭(72_1b)의 제1 로컬비트라인 센스앰프부(72_1a)에 구비된 로컬비트라인 센스앰프(72_1a_1)에 의해서 감지, 증폭된다.
- <143> 도10에 도시된 제2 로컬비트라인 센스앰프부(72_1c)는 종래의 비트라인 센스앰프부(도2의 123b 참조)와 같은 구성을 하고 있으나, 컬럼 제어신호(CD0,...)에 제어되어 비트라인(BL,/BL)에 인가된 데이터신호를 I/O 센스앰프부로 출력하는 데이터 출력부(도2의 123b_5 참조)가 없다.
- <144> 본 발명의 메모리 장치에서는 로컬비트라인 센스앰프(72_1c_3)에 의해 감지 증폭된 데이터신호는 일단 글로벌비트라인(74)을 통해 제1 및 제2 글로벌비트라인 센스앰프부(71a,71b)로 전달되어 래치되고, 이후에 제1 및 제2 글로벌비트라인 센스앰프부(71a,71b)에 래치된 데이터가 I/O 센스앰프부로 출력되기 때문에 로컬 비트라인 센스앰프부에는 데이터 출력부가 따로 필요없는 것이다.
- <145> 또한, 제2 기본셀블럭(72_2)의 제1 로컬비트라인 센스앰프(72_2a)는 로컬비트라인 연결부(72_2a)를 제외하고 나머지 회로는 도시하지 않았으나, 나머지 회로는 제1 기본셀블럭(72_1)의 제2 로컬비트라인 센스앰프부(72_1c)와 같은 구성을 가진다.
- <146> 본 발명의 메모리 장치에 구비된 단위셀블럭(예컨대 72_1b)에 1Mega의 단위셀을 구비하게 된다면 256개의 워드라인(WL0 ~ WL255)과, 4K개의 로컬비트라인쌍(BL0,/BL0 ~

BL1023 ×4, /BL1023 ×4)과, 2K개의 글로벌비트라인쌍(GBL0, /GBL0 ~ GBL1023 ×2, /GBL1023 ×2)과, 4K개의 로컬비트라인 센스앰프(4 ×024)를 구비하게 되는 것이다.

<147> 로컬비트라인쌍의 갯수가 4K이면 글로벌비트라인쌍의 갯수는 2K개이면되는데, 이는 셀블럭의 일측과 타측에 각각 2K개씩의 글로벌비트라인 센스앰프를 구비하고 단위셀블럭의 일측과 타측에 각각 구비된 2K개의 로컬비트라인 센스앰프에 각각 연결되기 때문이다. 즉, 한 워드라인에 대응하여 감지 증폭되어 4K개의 로컬비트라인 센스앰프에 래치된 데이터중 2K개의 제1 데이터는 글로벌비트라인의 일측으로 전달되고, 나머지 2K개의 데이터는 글로벌비트라인의 타측으로 전달되어 제1 및제2 글로벌비트라인 센스앰프부로 전달되기 때문이다. 글로벌비트라인(74)내에서의 데이터 충돌을 방지하기 위해 구비된 것이 글로벌비트라인 연결부(72_1 ~ 71_3)이다.

<148> 도11a 내지 도11d는 도8에 도시된 로컬비트라인 센스앰프(LSA)와 글로벌비트라인 센스앰프(GSA)간의 데이터 이동을 보여주는 도면이다. 특히 제1 글로벌비트라인(GBL0./GBL0)을 통해 두비트의 데이터가 양측에 구비된 글로벌비트라인 센스앰프(GSA, 71a_1, 71b_1)로 각각 이동되는 것을 나타낸 것이다.

<149> 로컬비트라인 센스앰프(LSA)에서 글로벌비트라인 센스앰프(GSA)로 데이터 이동은 로컬비트라인 센스앰프(LSA)에서 감지증폭된 데이터를 글로벌비트라인 센스앰프(GSA)로 래치하기 위한 것이고, 글로벌비트라인 센스앰프(GSA)에서 로컬비트라인 센스앰프(LSA)로의 데이터이동은 글로벌비트라인 센스앰프(GSA)에 래치된 데이터를 기본셀블럭에 재저장시키기 위해 이동시키는 것이다.

<150> 이하 도8 내지 도10과 도11a 내지 도11d를 참조하여 도8에 도시된 메모리 장치의 동작을 살펴본다.

- <151> 먼저 메모리 장치에 명령어가 입력되면, 그에 대응하여 입력된 어드레스에 의해 선택된 기본셀블럭에 저장된 데이터가 단위셀에 연결된 로컬비트라인 센스앰프(LSA)에 의해 감지 증폭된다. 이어서 로컬비트라인 센스앰프(LSA)에 의해 감지 증폭된 데이터는 글로벌비트라인 센스앰프(GSA)로 이동되어 래치된다.
- <152> 여기서 데이터는 한 비트의 데이터를 말하는 것이 아니고 입력된 어드레스에 의해 선택된 하나의 워드라인이 활성화되면, 활성화되는 워드라인에 대응하여 감지 증폭되는 모든 데이터를 말하는 것이다. 예를 들어 하나의 워드라인에 1024 × 4 개의 단위셀이 연결되어 있다면, 하나의 워드라인이 활성화되면 4K개의 단위셀에 각각 연결된 4K개의 로컬비트라인 센스앰프에 의해 4K개의 데이터가 감지 증폭되어서 글로벌비트라인 센스앰프로 이동되어 래치되는 것이다.
- <153> 도11a는 제2 기본셀블럭(72_2)의 로컬비트라인 센스앰프(LSA, 72_2a_1, 72_2c_1)에 의해 감지 증폭된 두비트의 데이터가 글로벌비트라인(GBL0, /GBL0)을 통해 글로벌비트라인 센스앰프(GSA, 71a_1, 71b_1)로 데이터가 각각 이동되어 래치되는 것을 나타내는 도면이다.
- <154> 도11a를 참조하여 살펴보면, 먼저 제1 명령어에 대응하여 입력된 어드레스에 의해 선택된 제2 기본셀블럭(72_2)에 저장된 데이터가 로컬비트라인 센스앰프(LSA, 72_2a_1, 72_2c_1)에 의해 감지, 증폭된다. 이어서 제2 기본셀블럭(72_2)의 로컬비트라인 센스앰프(LSA, 72_2a_1, 72_2c_1)에 의해 증폭된 데이터가 글로벌비트라인(GBL0, /GBL0)을 통해 글로벌비트라인 센스앰프(71a_1, 71b_1)로 이동되어 래치된다.
- <155> 이어서 글로벌비트라인 센스앰프(GSA)에 래치된 제1 데이터는 리드명령어에 대한 액세스동작에서는 출력데이터로 외부로 출력되고, 라이트명령어에 대한 액세스동작일 경

우에는 외부에서 입력된 입력데이터가 글로벌비트라인 센스앰프(GSA)에 래치된 제1 데이터를 덮어쓰게 된다.

<156> 이어서 다음실행될 제2 명령어에 대응하여 제2 기본셀블럭의 워드라인 하나가 선택되고, 선택된 워드라인에 대응하는 제2 데이터가 제2 기본셀블럭에 구비된 로컬비트라인 센스앰프(LSA)에 의해 감지 증폭된다.

<157> 한편, 제2 명령어에 대응하여 데이터가 감지 증폭되는 타이밍에 글로벌비트라인 센스앰프(GSA)에 래치된 제1 데이터는 제5 기본셀블럭의 로컬비트라인 센스앰프(LSA)로 이동되어 래치되고, 이어서 제1 명령어에 대응하는 재저장동작이 제5 기본셀블럭에서 이루어진다.

<158> 도11b는 글로벌비트라인 센스앰프(71a_1, 71b_1)에서 제5 기본셀블럭(72_5)으로 글로벌비트라인(GBL, /GBL)을 통해 데이터가 이동되는 것이 도시되어 있다.

<159> 도11b를 참조하여 살펴보면, 글로벌비트라인 센스앰프(GSA, 71a_1, 71b_1)에 저장되어 있던 데이터가 제5 기본셀블럭(72_5)의 로컬비트라인 센스앰프(72_5a_1, 72_5c_1)으로 이동되고, 제5 기본셀블럭(72_5)에서 제1 명령어에 대응하는 재저장동작이 이루어진다.

<160> 여기서 제1 명령어에 대응하는 재저장동작과 제2 명령어에 대응하는 데이터의 감지 증폭 동작은 실질적으로 동일한 타이밍에 이루어진다. 이는 제1 명령어와 제2 명령어가 연속해서 동일한 제2 기본셀블럭을 액세스하게 되지만 제1 명령어에 대응하는 재저장 동작은 제2 기본셀블럭에서 이루어지는 것이 아니고 제5 기본셀블럭에서 이루어지기 때문에 가능한 것이다.

- <161> 또한, 전술한 바와 같이 본 발명의 메모리 장치는 하나의 선택된 단위셀에 저장된 데이터가 감지 증폭되면 일단 글로벌비트라인 센스앰프로 이동되어 래치되고, 래치된 데이터를 이용하여 실행명령어에 대응하는 출력데이터를 출력하게 된다.
- <162> 또한, 본 발명의 메모리 장치는 제1 기본셀블럭(72_1)과 제2 기본셀블럭(72_2)에서 제1 데이터 및 제2 데이터가 교대로 액세스 될 때에는 제1 기본셀블럭(72_1)에서 제1 데이터를 재저장하는 타이밍과 실질적으로 동일한 타이밍에 제2 기본셀블럭(72_2)에서 상기 제2 데이터가 감지 증폭되도록 한다.
- <163> 11c와 11d는 제4 기본셀블럭에서 데이터를 감지 증폭하여 글로벌비트라인 센스앰프에 래치시키고, 이어서 글로벌비트라인 센스앰프에서 제1 기본셀블럭로 데이터가 이동되어 재저장되는 것을 나타내고 있다. 여기서의 동작은 도11a 및 도11b에서 설명한 동작과 같으므로 자세한 설명은 생략한다.
- <164> 한편, 종래에는 메모리 장치의 면적감소를 위해 하나의 로컬센스앰프를 이웃한 셀블럭이 공유해서 사용하였다.
- <165> 이에 반하여, 본 발명은 워드라인 하나의 대응되는 모든 데이터를 감지 증폭할 수 있는 로컬비트라인 센스앰프를 기본셀블럭의 일측과 타측에 모두 구비하고, 셀블럭 영역의 일측과 타측에 로컬비트라인 센스앰프에서 감지 증폭된 데이터를 래치할 수 있는 글로벌 비트라인 센스앰프를 구비하고 있는 것이 특징이다.
- <166> 또한 기본셀블럭의 로컬비트라인 센스앰프와 글로벌 비트라인 센스앰프간의 데이터 이동을 위한 글로벌 비트라인을 추가로 구비하고, 이웃한 2개의 로컬비트라인 센스앰프

를 글로벌 비트라인과 선택적으로 연결하기 위한 글로벌비트라인 연결부를 2개의 기본셀블럭당 하나씩 구비하고 있다.

<167> 따라서 종래의 메모리 장치보다 셀영역의 면적이 증가되는 부분이 있으나 상기와 같이 셀영역을 구성하게 되면, 하나의 기본셀블럭에 연속해서 데이터가 액세스될 때에도 데이터를 다시 원래의 기본셀블럭에 재저장하지 않아도 되기 때문에 연속해서 다음 데이터를 액세스할 수 있어 고속으로 데이터를 액세스할 수 있게 된다.

<168> 한편, 도8에 도시된 바와 같이 기본셀블럭의 일측과 타측에 글로벌비트라인 센스앰프부를 구비하게 되는 것은 기본셀블럭의 구성이 구비된 선택된 워드라인에 대응하는 데이터(예컨대 4K개의 데이터)중에서 일부 데이터(2K개의 데이터)는 일측에 구비된 로컬비트라인 센스앰프부에 의해 감지 증폭되고, 나머지 데이터(2K개의 데이터)는 타측에 구비된 로컬비트라인 센스앰프부에 의해 감지 증폭되는 구조이기 때문이다.

<169> 즉, 2개의 글로벌비트라인 센스앰프부는 일측과 타측에 구비된 로컬비트라인 센스앰프부에 의해 감지 증폭된 데이터를 각각 이동시켜 래치하기 위한 것이다.

<170> 따라서 기본셀블럭이 하나의 활성화된 워드라인에 대응하는 모든 데이터가 일측에 구비된 로컬비트라인 센스앰프에 의해 감지 증폭되는 구조라면 글로벌 비트라인 센스앰프부는 일측에만 구비하면 된다. 또한 이 경우에는 글로벌 비트라인 연결부를 따로 필요하지 않게 되는 것이다.

<171> 도12는 하나의 글로벌비트라인 센스앰프부만을 구비하여 고속으로 데이터 액세스가 가능하도록 구성된 본 발명의 메모리 장치의 블럭구성도이다.

- <172> 도12를 참조하여 살펴보면, 본 발명의 메모리 장치는 다수의 단위셀을 각각 구비하고, 상기 다수의 단위셀에 저장된 데이터를 감지증폭하기 위한 로컬비트라인 센스앰프부를 각각 구비하는 다수의 기본셀블럭(62_1',62_2,62_3)와, 기본셀블럭(62_1,62_2,62_3)에 각각 구비된 다수의 로컬비트라인 센스앰프부(62_1b,62_2b,62_3b)에 의해 감지 증폭된 데이터를 래치하기 위한 글로벌비트라인 센스앰프부(62)와, 래치된 데이터를 재저장 동작에 사용되도록 제어하기 위한 제어부(50)를 구비한다. 도면부호 BL,/BL은 단위셀블럭(61_1a)에 구비된 로컬비트라인을 나타낸다.
- <173> 또한 다수의 로컬비트라인 센스앰프부(62_1b,62_2b,62_3b)는 상기 글로벌비트라인 센스앰프부와 선택적으로 연결되기 위한 스위치(도10의 72_1c_4)를 각각 구비한다.
- <174> 또한 다수의 로컬비트라인 센스앰프부(62_1b,62_2b,62_3b)와 글로벌비트라인 센스앰프부(62)를 연결하기 위한 글로벌 비트라인(63)을 더 구비한다.
- <175> 도13은 도12의 기본셀블럭의 일부분을 나타내는 회로도로서, 특히 제1 기본셀블럭(61_1)의 단위셀블럭(61_1a)과 로컬비트라인 센스앰프부(61_1b)를 나타낸 것이다.
- <176> 도13에 도시된 회로구성은 도10에 도시된 회로구성과 기본적으로 같으나, 한 워드라인에 대응하는 모든 데이터를 감지 증폭하는 로컬비트라인 센스앰프(61_1b_3)가 일측에 모두 구비되어 있기 때문에 로컬비트라인(BL0,/BL0, ...)과 글로벌비트라인(GBL0./GBL)을 연결하는 글로벌 비트라인 연결부(도10의 73_1)가 따로 필요없다. 이 경우에는 로컬비트라인에 구비된 스위치부(61_1b_4)를 이용하여 로컬비트라인(61_1b_3)을 글로벌비트라인(GBL0./GBL)을 선택적으로 연결하게 된다.

- <177> 또한, 로컬비트라인 센스앰프(61_1b_3)는 도10에 도시된 로컬비트라인 센스앰프(LSA)와 같은 회로구성을 가지며, 프리차지부(61_1b_2)는 로컬비트라인 센스앰프가 디스에이블되었을 때에 비트라인쌍(BL0, /BL0)의 전압레벨을 프리차지전압(Vb1p)로 프리차지시키는, 역할을 하고, 이퀄라이제이션부(61_1b_2)는 비트라인쌍(BL0, /BL0)의 전압을 같게 유지시키는 역할을 한다.
- <178> 도12에 도시된 메모리 장치의 동작은 도8에 도시된 메모리 장치의 동작과 유사하므로 자세한 동작설명은 생략한다.
- <179> 한편, 도12에 도시된 메모리 장치에서 글로벌비트라인 센스앰프부를 일측과 타측에 각각 구비할 수도 있는데, 이 경우에는 일측과 타측에 구비된 글로벌비트라인 센스앰프부는 각각 2K개의 글로벌비트라인과 연결되고, 한 단위셀블럭에서 감지 증폭되어 로컬비트라인에 센스앰프에 래치된 4K개의 데이터는 각각 셀블럭의 일측과 타측에 구비된 2K개씩 글로벌비트라인 센스앰프부로 이동되어 래치된다.
- <180> 이렇게 셀블럭을 구성하게 되면, 글로벌 비트라인 4개당 하나의 글로벌비트라인 센스앰프를 구현할 수 있게 되어 셀블럭의 레이아웃 영역에 따라서 효율적일 수 있으며, 도8에 도시된 글로벌비트라인 연결부는 필요없게 된다.
- <181> 또한, 도8에 도시된 메모리 장치에서는 셀블럭에 구비된 글로벌비트라인의 수를 4K로 구성하여, 한 단위셀블럭에 구비된 2K개의 로컬비트라인센스앰프와 제1 및 제2 글로벌비트라인 센스앰프부를 각각 글로벌비트라인으로 연결시키게 되면 글로벌비트라인 연결부가 필요없게 된다.

- <182> 전술한 바와 같이 메모리 장치에서와 같이 데이터를 액세스한 셀블럭이 아닌 다른 셀블럭에 상기의 데이터를 제저장하는 경우에는 입력되는 어드레스에 대응하는 셀블럭외에도 여분의 셀블럭이 필요하게 된다.
- <183> 도14는 본 발명의 메모리 장치를 나타내는 블럭구성도로서, 입력되는 어드레스에 대응하는 셀블럭과 데이터 제저장을 위한 여분의 셀블럭을 가지는 메모리 장치의 블럭구성도이다.
- <184> 도14를 참조하여 살펴보면, 본 발명의 메모리 장치는 각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 N개로 구성된 단위셀블럭(40_1 ~ 40_n)에, 추가적으로 M개의 워드라인을 가지는 단위셀블럭(40_{n+1})을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭(40)과, N+1개의 단위셀블럭(40_1 ~ 40_n, 40_{n+1})중에서 선택된 하나의 단위셀블럭(예컨대 40_1)으로부터 액세스되는 데이터를 선택된 하나의 단위셀블럭(40_1) 또는 다른 단위셀블럭(예컨대 40_3)으로 제저장시키도록 제어하는 제어부(30)을 구비한다.
- <185> 제어부(30)는 입력되는 로우어드레스에 대응하여 2개의 단위셀블럭이 활성화되도록 제어하게 된다.
- <186> 또한 본 발명의 메모리 장치는 N+1개의 단위셀블럭(40_1 ~ 40_n, 40_{n+1})중에서 선택된 하나의 단위셀블럭(예컨대 40_1)에서 활성화되는 워드라인(WL0)과 다른 단위셀블럭(예컨대 40_3)에서 활성화되는 워드라인(WL0)을 선택하기 위한 각각의 어드레스는 동일 어드레스를 사용한다.

- <187> 즉, 제어부(30)에 의해 활성화되는 2개의 단위셀블럭에서는 같은 어드레스를 가지는 2개의 워드라인을 활성화시키는 것인데, 예를 들어 제어부에서 입력된 어드레스에 의해 제1 단위셀블럭(40_1)의 워드라인 'WL0'이 활성화 되면, 제3 단위셀블럭(40_3)에서도 워드라인 'WL0'을 활성화시키는 것이다.
- <188> 제어부(30)는 N+1개의 단위셀블럭(40_1 ~ 40_n, 40_n+1)중에서 선택된 제1 단위셀블럭(40_1)에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 제1 단위셀블럭(40_1)에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 N+1단위셀블럭중에서 선택된 다른 단위셀블럭에서는 상기 제1 데이터의 재저장 동작이 수행하도록 제어한다.
- <189> 도15는 도14에 도시된 메모리 장치의 동작을 나타내는 도면으로서, 도14에 도시된 N+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭에 제1 데이터와 제2 데이터가 연속적으로 액세스되는 경우에 동작타이밍을 나타내는 도면이다.
- <190> 이하 도14 및 도15을 참조하여 도14에 도시된 본 발명의 메모리 장치의 동작에 대해서 설명한다.
- <191> 먼저 본 발명의 메모리 장치는 입력되는 어드레스에 대응하는 단위셀블럭이 N개라면 하나의 단위셀블럭을 더 구비하여 총 N+1개의 단위셀블럭을 구비한다. 예를 들어 입력되는 어드레스에 대응하는 단위셀블럭이 8개라면 총9개의 단위셀블럭을 구비하는 것이다.
- <192> 추가적으로 구비되는 단위셀블럭은 동일한 단위셀블럭에 연속해서 데이터 액세스가 일어날 때 이전 명령어에 의해 액세스된 데이터 재저장을 위한 것이다.

- <193> 도15에서는 N+1개의 단위셀블럭중에서 제1 단위셀블럭에 연속적으로 데이터가 역세스되고, 입력되는 어드레스에 의해 워드라인 WL0, WL1이 연속해서 활성화되는 것으로 가정하였다.
- <194> 도15를 참조하여 살펴보면, 먼저 제1 명령어(CD0)에 의해 N+1개의 단위셀블럭 중에서 선택된 제1 단위셀블럭(40_1)의 제1 워드라인(WL0)을 활성화시킨다. 이어서, 제1 워드라인(WL0)에 대응되는 K개의 데이터를 감지 증폭한다. 이어서 제1 워드라인(WL0)에 대응하는 K개의 데이터를 데이터래치부(20)로 이동시켜 래치시킨다. 이어서, 제1 워드라인(WL0)에 대응하여 K개의 데이터를 감지, 증폭한 센스앰프를 강제프리차지시킨다.
- <195> 이어서 제1 명령어(CD0)가 리드명령어인 경우에는 데이터래치부(20)에 래치된 K개의 데이터중 선택된 하나의 데이터를 제1 명령어(CD0)에 대응하는 출력데이터로 출력시키고, 제1 명령어(CD))가 라이트명령어인 경우에는 입력되는 데이터로 데이터래치부(20)에 래치된 K개의 데이터중 선택된 하나의 데이터를 덮어쓰게 된다.
- <196> 이어서 제1 워드라인(WL0)에 대응하는 예비워드라인으로 지정된 제3 단위셀블럭(40_3)의 워드라인(WL0)을 활성화시킨다. 이어서 데이터래치부(20)에 래치된 K개의 데이터를 제3 단위셀블럭(40_3)의 워드라인(WL0)에 대응되는 K개의 단위셀에 재저장한다.
- <197> 이어서 제2 명령어(CD1)에 대응하여 선택된 제1 단위셀블럭(40_1)에 제2 워드라인(WL1)을 활성화시킨다. 이어서 제2 워드라인(WL1)에 대응하는 K개의 데이터를 감지증폭한다.

- <198> 여기서 제1 명령어(CD0)에 대응하여 감지, 증폭된 K개의 데이터가 재저장 동작과 제2 명령어(CD1)에 의해 K개의 데이터가 감지, 증폭 동작은 실질적으로 같은 타이밍에 이루어진다.
- <199> 이렇게 데이터를 액세스 함으로서 이전명령어에 의해 감지, 증폭된 K개의 데이터가 재저장되는 타이밍에 상관없이 다음 명령어에 의해 데이터를 감지, 증폭할 수 있게 되어, 데이터 재저장시간만큼 데이터엑세스 시간을 줄일 수 있게 된다.
- <200> 한편 입력되는 명령어에 의해 활성화되는 워드라인에 대해서는 하나의 예비워드라인이 필요하게 되는데, 예비 워드라인은 총 M개가 지정되며, 하나에 단위셀블럭 구비된 모든 워드라인이 예비워드라인이 되는 것이 아니고, 메모리 장치의 동작중에 계속 바뀌게 되는 것이다. 이는 어드레스에 대응하는 단위셀블럭보다 추가적으로 하나의 단위셀블럭을 구비함으로서 가능하게 된다.
- <201> 예컨대 제1 단위셀블럭(40_1)의 워드라인(WL0)에 대한 예비워드라인은 제3 단위셀블럭의 워드라인(WL0)이고, 제1 단위셀블럭(40_1)의 워드라인(WL1)에 대한 예비워드라인은 제2 단위셀블럭(40_2)의 워드라인(WL1)으로 지정될 수 있는 것이다.
- <202> 이어서 제1 데이터 및 제2 데이터가 서로 다른 단위셀블럭에서 교대로 액세스되는 인터리빙모드로 동작하는 것에 대해서 살펴본다.
- <203> 먼저 상기 제1 데이터를 상기 N+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭(40_1)에서 감지 증폭한 다음 데이터래치부에 래치시킨다. 이어서 제1 단위셀블럭(40_1)에 제1 데이터를 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 상기

N+1개의 단위셀블럭중에서 선택된 제2 단위셀블럭(40_2)에서 상기 제2 데이터를 감지 증폭하여 상기 데이터래치부에 래치시킨다.

<204> 따라서 본 발명의 메모리장치는 데이터가 하나의 셀에 계속해서 액세스되는 경우에는 이전 데이터에 대한 재저장은 지정된 예비워드라인에 대응하는 단위셀에서 수행하고, 데이터가 단위셀에 교대로 액세스되는 경우에는 인터리빙모드로 동작함으로써, 액세스되는 패턴에 상관없이 고속으로 데이터를 액세스 할 수 있게 되는 것이다.

<205> 한편, 입력되는 어드레스에 대응하는 단위셀블럭의 수가 N개이고, 실제 단위셀블럭은 N+1개를 구비하기 위해서는 N개에 대응하여 입력되는 논리적 셀블럭어드레스를 N+1개에 대응하는 물리적 셀블럭어드레스로 변환하는 변환부가 필요하다.

<206> 또한, 선택된 하나의 단위셀블럭에 데이터가 연속적으로 액세스될 때에 이전 데이터에 대한 재저장을 다른셀블럭(액세스되는 워드라인에 대해 지정된 예비워드라인이 있는 셀블럭)에서 수행하기 위해서는 현재 활성화된 워드라인에 대한 예비워드라인이 어떤 단위셀블럭에 있는지를 저장하고 있는 태그저장부가 필요하다.

<207> 도16은 본 발명에 의해서 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하기 위한 태그블럭(430)과, 예비워드라인에 대한 정보를 저장하고 있는 예비셀블럭 테이블(410)을 구비하는 메모리 장치의 블럭구성도이다.

<208> 특히 도16은 셀블럭(500)이 9개의 단위셀블럭을 구비하고, 하나의 단위셀블럭에 구비된 워드라인이 256개인 경우를 도시한 것으로, 이하에서는 본 발명의 메모리 장치가 상기와 같이 구비한 것으로 가정하고 설명한다.

<209> 도16을 참조하여 살펴보면, 본 발명의 메모리 장치는 각각 256개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 8개의 단위셀블럭에, 추가적으로 256개의 워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 9개의 단위셀블럭으로 구성되는 셀블럭(500)과, 셀블럭에(500) 구비되는 $(8+1) \times 256$ 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블(410)과, 상기 로우어드레스를 입력받아 N개의 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 상기 9개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하기 위한 태그블럭(430)과, 상기 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 상기 워드라인에 대한 -상기 예비셀블럭 테이블로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 상기 태그블럭과 상기 예비셀블럭 테이블을 제어하는 제어부(420)을 구비한다.

<210> 또한 제어부(420)는 상기 9개의 단위셀블럭중에서 선택된 하나의 단위셀블럭에 연속적으로 제1 및 제2 데이터가 액세스될 때, 상기 제1 데이터에 대한 재저장동작은 상기 제2 데이터를 액세스하기 위해 활성화된 워드라인에 대응하여 지정된 예비워드라인을 통해 이루어지도록 제어한다.

<211> 또한 본 발명의 메모리 장치에 입력되는 어드레스에는 8×256 개의 워드라인에 대응하며, 추가의 단위셀블럭의 구비되는 워드라인의 수만큼인 256개를 예비워드라인으로 지정한다. 그러나 예비워드라인은 한 단위셀블럭에 있는 256개의 워드라인으로 정해지는 것은 아니고 메모리 장치가 동작중에서 계속해서 위치가 변하게 되는데, 이에 대해서는 뒤에서 자세히 설명한다.

- <212> 또한 본 발명의 메모리 장치에 입력되는 로우어드레스는 8개의 단위셀블럭중 하나를 선택하기 위한 논리적셀블럭어드레스와, 선택된 하나의 단위셀블럭에 구비된 256개의 워드라인중 하나를 선택할 수 있는 로컬어드레스를 포함한다.
- <213> 또한 셀블럭(500)에 구비되는 9개의 단위셀블럭은 액세스되는 데이터를 감지증폭하기 위한 로컬비트라인 센스앰프(미도시, 도8참조)를 각각 구비하며, 셀블럭(500)은 상기 로컬비트라인 센스앰프에서 감지증폭되는 데이터를 래치하기 위한 데이터래치부(450)를 더 구비한다.
- <214> 도17은 도16에 도시된 태그블럭(430)을 나타내는 블럭구성도이다.
- <215> 도17을 참조하여 살펴보면, 태그블럭(430)은 9개의 단위셀블럭에 각각 구비된 256개의 워드라인이 어떤 논리적 셀블럭에 대응되는 것인지를 각각 저장하고 있는 9개의 단위태그테이블(432a~432i)와, 입력되는 논리적셀블럭어드레스(Cur_LBA)와, 입력되는 로컬어드레스(Cur_RA)에 대응되어 상기 9개의 단위태그테이블(432a~432i)에서 출력되는 정보-상기 로컬어드레스(Cur_RA)에 의해 선택된 워드라인이 어떤 논리적셀블럭에 관한 것에 관한 9개의 데이터 정보-와 각각 비교하기 위한 9개의 비교부(433a~433i)와, 9개의 비교부(433a~433i)에서 비교한 정보를 인코딩하여 물리적 셀블럭어드레스(Cur_PBA)를 출력하기 위한 인코더(434)와, 9개의 단위태그테이블(432a~432i)과, 9개의 비교부(433a~433i)와 인코더(434)를 제어하기 위한 태그제어부(436)를 구비한다.
- <216> 태그블럭(430)은 예비셀블럭 테이블(420)에서 출력되는 정보(Extra_BA) -로컬어드레스(Cur_RA)에 대응하는 워드라인의 예비워드라인이 9개의 단위셀블럭(500)중에서 어떤 단위셀블럭에 있는지에 대한 정보-를 디코딩하여, 9개의 단위태그테이블(432a~432i)중에서 하나의 단위태그테이블을 선택하기 위한 신호를 출력하는 셀블럭어드레스

디코더부(431)를 더 구비한다. 또한 현재 동작클럭(CK)의 물리적 셀블럭어드레스 (Cur_PBA)를 한클럭 지연시켜 이전 클럭의 물리적 셀블럭어드레스(Pre_PBA)를 출력하기 위한 태그지연부(435)을 구비한다.

<217> 여기서 각각의 단위태그테이블(432a~432i)은 256개의 레지스터를 구비하고, 논리적 셀블럭어드레스가 8개이므로 각각의 레지스터는 3비트로 구성된다. 예를 들어 제1 단위 태그테이블(432a)는 제1 단위셀블럭에 구비된 256개의 워드라인이 어떤 논리적셀블럭에 대한 것인지를 저장하고 있고, 제2 단위태그테이블(432b)는 제2 단위셀블럭에 구비된 256개의 워드라인이 각각 어떤 논리적셀블럭에 대한 것인지를 저장하고 있는 것이다.

<218> 또한, 각각의 태그메모리(432a~432i)에서 제1 레지스터(0)는 워드라인 'WL0'의 논리적 셀블럭어드레스를 저장하고 제2 레지스터(1)는 워드라인 'WL1'의 논리적 셀블럭어드레스를 저장하고, 제256 레지스터(255)는 워드라인 'WL255'의 논리적 셀블럭어드레스를 저장한다.

<219> 예컨대 도17에 도시된 제1 단위태그테이블(432a)을 참조하여 살펴보면, 제1 단위셀블럭의 워드라인(WL0)(제1 레지스터 '0'참조)에 대응하여 저장된 데이터는 제2 논리적 단위셀블럭의 워드라인(WL0)에 대응하는 데이터이고, 제1 단위셀블럭의 워드라인(WL255)(제255 레지스터 '255'참조)에 대응하여 저장된 데이터는 제7 논리적 단위셀블럭의 워드라인에 대응하는 데이터인 것이다.

<220> 태그제어부(436)는 셀블럭어드레스 디코더부(431)의 출력 -선택된 워드라인에 대한 예비 워드라인이 어떤 물리적셀블럭에서 지정되었는지에 대한 정보-에 의해 선택된 단위태그테이블의 상기 워드라인에 의해 대응되는 레지스터에 논리적 셀블럭어드레스 (Cur_LBA) 정보를 저장하도록 9개의 단위태그테이블(432a~432i)을 제어한다.

- <221> 예를 들어, 현재 입력된 논리적 셀블럭어드레스(Cur_LBA)와 로컬어드레스(Cur_RA)가 각각 제3 단위셀블럭과 워드라인 'WL1'을 선택하고, 선택된 워드라인(WL1)에 대응하는 예비워드라인이 제1 단위셀블럭의 'WL1'로 지정된 경우에, 태그제어부(436)의 제어에 의해서 제1 단위태그테이블(432a)의 두번째 레지스터(1)에 논리적 셀블럭어드레스 (Cur_LBA:3)을 저장되는 것이다.
- <222> 도18은 도16의 예비셀블럭 테이블(410)의 블럭구성도이다.
- <223> 도18을 참조하여 살펴보면, 예비셀블럭 테이블(410)은 256개의 예비워드라인이 9개의 물리적 단위셀블럭중에서 어떤 단위셀블럭에 있는지에 대한 정보를 저장하기 위한 256개의 레지스터(411)를 구비한다. 하나의 단위셀블럭에 총 256개의 워드라인이 있기 때문에, 256개의 워드라인(WL)에 대한 예비워드라인 정보를 저장하기 위해 예비셀블럭 테이블(410)에는 256개의 레지스터가 있고, 각각의 레지스터는 9개의 단위셀블럭에 대한 정보를 저장해야 하기 때문에 각각 4비트로 구현된다.
- <224> 예비셀블럭 테이블(410)에 저장된 내용을 살펴보면, 워드라인(WL0)의 예비워드라인은 제2 단위셀블럭의 워드라인 'WL0'이라는 것이고(제1 레지스터'0' 참조), 워드라인(WL3)의 예비워드라인은 제4 단위셀블럭의 워드라인 'WL3'이라는 뜻이다(제4 레지스터 '3' 참조). 메모리 장치의 동작중에 레지스터(411)에 저장된 내용은 계속 업데이트되며, 업데이트 될 때마다 256개의 예비워드라인에 대한 정보는 계속 바뀌는 것이다.
- <225> 또한, 예비셀블럭 테이블(410)은 현재의 로컬어드레스(Cur_LA)를 입력받아 현재 실행되고 있는 로컬어드레스(Cur_LA)에 대한 예비워드라인이 어떤 단위셀블럭에 지정되었는지를 알려주는 신호(Extra_BA)를 태그블럭(430)으로 출력하고, 이전 단계의 로컬어드

레스(Pre_RA)와, 이전단계의 물리적 셀블럭어드레스(Pre_PBA)를 입력받아 예비블럭 업데이터 신호(EBT_UPDATE)에 응답하여 레지스터(411)을 업데이터한다.

<226> 도19는 도16에 도시된 제어부(420)의 일부를 나타내는 블럭구성도이다.

<227> 도19를 참조하여 살펴보면, 제어부(420)는 외부에서 입력되는 명령어(Ext_CMD)/논리적 셀블럭어드레스(Ext_BA)/로컬어드레스(Ext_RA)를 입력받아 현재 동작클럭에서의 명령어(Cur_CMD)/논리적 셀블럭어드레스(Cur_LBA)/로컬어드레스(Cur_RA)와, 이전 동작클럭에서의 명령어(Pre_CMD)/논리적 셀블럭어드레스(Pre_LBA)/로컬어드레스(Pre_RA)를 출력하기 위한 신호입력부(421)를 구비한다.

<228> 또한 제어부(420)는 현재 클럭에서의 실행 명령어(Cur_CMD)/논리적 셀블럭어드레스(Cur_LBA)/로우어드레스(Cur_RA)와, 이전 클럭에서의 실행 명령어(Pre_CMD)/논리적 셀블럭어드레스(Pre_LBA)/로우어드레스(Pre_RA)를 입력받는 어드레스 변환제어부(422)를 구비하는데, 어드레스 변환 제어부(422)에서 출력되는 예비셀블럭 업데이터 신호(EBT_UPDATE)는 이전 클럭과 현재클럭에 액세스할 물리적 셀블럭어드레스(Cur_PBA, Pre_PBA(이 신호는 제어부(422)내부에서 생성된다))가 같을 경우, 예비셀블럭 테이블(410)에 저장된 내용을 업데이터하기 위한 신호이다.

<229> 프리차지 액티브신호(pc_act[0:8])는 입력되는 명령어에 대응하는 물리적 셀블럭어드레스를 선택하기 위한 신호이다.

<230> 또한 내부 재저장신호(intW[0:8])는 이전 클럭과 현재클럭에 액세스할 물리적 셀블럭어드레스(Cur_PBA,Pre_PBA(이 신호는 제어부(422)내에서 생성된다))가 같을 경우, 현재 실행되는 명령어에 의해 선택된 단위셀블럭에서 액세스되어 데이터래치부에 래치된

데이터를 예비 워드라인이 지정된 단위셀블럭으로 이동시켜 재저장하도록 하는 신호이다.

- <231> 클럭지연부(422_1)는 내부 재저장신호(intW[0:8])를 프리차지 액티브신호(pc_act[0:8])에 비해 한클럭 지연되어 출력하기 위한 것이다. 이는 먼저 프리차지 신호(pc_act[0:8])를 이용하여 단위셀블럭을 선택하고, 선택된 단위셀블럭중 하나의 워드라인을 선택하고, 선택된 워드라인에 대응되는 데이터를 감지 증폭한 다음, 내부 재저장신호(intW[0:8])를 이용하여 데이터의 재저장을 실행하기 위한 것이다.
- <232> 또한, 명령어 취소신호(CMD_KILL)는 연속해서 동일한 셀블럭의 동일 워드라인을 액세스하려고 할 때에는 뒤에 실행되는 명령어의 동작을 중지시키기 위한 신호이다. 이는 연속해서 같은 셀블럭의 같은 워드라인을 액세스하는 경우에, 앞 명령어에 대해 태그블럭(430)에서 논리적 셀블럭어드레스에 대한 물리적 셀블럭어드레스에 대한 정보가 제대로 업데이트 되지 않은 상태에서 태그 정보를 읽게되면 오류가 생기기 때문이다. 이 때에는 나중의 명령어에 관한 동작을 취소시키고, 이전의 데이터를 그대로 액세스하면 되는 것이다.
- <233> 도20은 도17에 도시된 9개의 단위태그테이블(432a~432i)의 일예를 나타내는 회로도이고, 도21은 도18에 도시된 예비셀블럭테이블(410)의 일예를 나타내는 회로도이다.
- <234> 도20에 도시된 바와 같이, 하나의 단위태그테이블(예컨대 432a)은 통상적인 메모리 장치의 단위셀블럭에 사용되는 단위셀을 이용하여 구성할 수 있는데, 각각의 레지스터는 8개의 단위셀블럭에 대응하는 논리적 셀블럭어드레스를 저장하기 때문에 3비트씩이 필요하고, 하나의 단위셀블럭에 256개의 워드라인이 구비되기 때문에 하나의 단위태그메

모리는 256개의 레지스터가 필요하게 된다. 따라서 태그블럭에 필요한 메모리의 총크기는 $256 \times 8 \times 9$ 비트가 된다.

<235> 한편, 예비셀블럭 테이블(410)도 도20에 도시된 바와 같은 단위셀블럭의 단위셀 구조를 이용해서 구성할 수 있으나, 예비블럭테이블(410)은 한 클럭내에서 정보를 읽고, 다시 업데이트를 해야하기 때문에 도21에 도시된 바와 같이 고속억세스가 가능한 스택 셀로 구성하는 것이 바람직하다.

<236> 예비셀블럭 테이블은 256개의 예비워드라인을 지정하기 위해 256개의 레지스터가 필요하면, 하나의 레지스터는 9개의 단위셀블럭에 대한 정보를 저장해야 하기 때문에 4비트가 필요하다. 따라서 예비셀블럭에 필요한 메모리의 총 크기는 256×4 비트가 된다.

<237> 도22은 본 발명에 의한 메모리 장치의 동작타이밍도로서, 특히 태그관련 블럭의 동작을 중심으로 나타내는 것이다.

<238> 이하에서는 도16 내지 도22을 참조하여 본 발명에 따른 메모리 장치의 동작에 대해서 설명한다.

<239> 본 발명에 의한 메모리 장치의 동작은 한 타이밍(t_0, t_1, \dots)에 하나의 명령어를 입력받으며, 한 명령어가 실행되기 위해서는 2개의 타이밍이 필요하다. 예를 들면 제1 명령어(CD0)에 따른 동작은 첫번째 타이밍과 두번째 타이밍(t_0, t_1) 동안 실행되는 것이다. 한번의 타이밍(t_0, t_1, \dots)에는 각각 태그관련 블럭(410, 430등)이 동작하는 구간(a_0, a_1, \dots)과 셀블럭(500)이 동작하는 구간(b_0, b_1, \dots)으로 구분된다. 이하에서 각각의 구간별로 메모리 장치가 수행하는 동작을 자세히 설명한다.

- <240> 먼저 첫번째 타이밍(t_0)에서의 태그관련 블록이 동작하는 구간(a_0 구간)을 살펴본다.
- <241> a_0 구간의 첫번째 구간인 a_0-1 구간에서는 제1 명령어(CD0)에 대응하여 로우어드레스 (Row Address)가 입력되면, 제어부(420)에서는 로우어드레스를 셀블럭(500)에 구비된 8개의 단위셀블럭중에서 하나를 선택할 수 있는 논리적 셀블럭어드레스와, 한 단위셀블럭에 구비된 256개의 워드라인중 하나를 선택할 수 있는 로컬어드레스로 분리하여 입력받는다. (a_0-1 구간)
- <242> 이어서 a_0-2 구간에서는 태그블럭(430)에서 논리적 셀블럭어드레스(Cur_LBA)에 해당되는 데이터가 9개의 물리적셀블럭중 어디에 있는지를 나타내는 물리적 셀블럭어드레스 (Cur_PBA)로 변환한다. (a_0-2 구간)
- <243> 태그블럭(430)에서의 어드레스 변환과정(a_0-2 구간)을 자세히 살펴보면, 먼저 9개의 단위태그테이블(432a~432i)에서는 현재 실행되는 명령에 따른 로컬어드레스(Cur_RA)에 대응하는 레지스터에 저장된 9개의 논리적 셀블럭어드레스를 비교부(433a~433i)로 출력한다. 예를 들어 로컬어드레스(Cur_RA)가 '1'이면, 제1 단위태그테이블(432a)에서는 제2 레지스터(1)에 저장된 논리적셀블럭어드레스 '2'를 출력하고, 제2 단위태그테이블(432b)는 제2 레지스터(1)에 저장된 논리적셀블럭어드레스 '1'을 비교부(433a, 433b)로 각각 출력하는 것이다.
- <244> 이어서 9개의 비교부(433a~433i)에서는 현재 실행되는 명령에 따른 논리적 셀블럭 어드레스(Cur_LBA)와 9개의 단위태그테이블(432a~432i)에서 출력되는 9개의 논리적 셀블럭 어드레스를 각각 일치하는지를 비교한다. 예를 들어 로컬어드레스(Cur_RA)이 '1'이고

논리적 셀블럭어드레스(Cur_LBA)가 '2'라면, 비교부(433a)에서 현재 실행되는 명령에 따른 논리적 셀블럭어드레스(Cur_LBA) '2'와 단위태그테이블(432a)에서 출력되는 데이터 '2'가 일치되는 것이다.

<245> 이어서 셀블럭어드레스 인코더부(434)에서는 9개의 비교부(433a~433i)에서 출력되는 정보를 인코딩하여 현재 실행명령의 물리적 셀블럭어드레스(Cur_PBA)를 출력한다.

<246> 이어서 태그지연부(435)에서는 클럭신호를 입력받아 현재 실행명령에 따른 물리적 셀블럭어드레스(Cur_PBA)를 한클럭 지연시킨 이전명령의 물리적 셀블럭어드레스(Pre_PBA)를 출력하는 것이다.

<247> 한편, 계속해서 a_0 -3구간에 대해 살펴보면, 예비블럭테이블(410)에서 현재 입력된 로컬어드레스(Cur_RA)에 해당되는 워드라인(예컨대 WL0)의 예비워드라인이 어떤 단위셀블럭에 있는지 감지하여 예비워드라인이 있는 셀블럭어드레스(Extra_BA)를 제어부(420)와 태그블럭(430)으로 출력한다.

<248> 예를 들면, 로컬어드레스가 '1'인 경우에는 워드라인은 'WL1'이 선택되어 활성화될 것이고, 예비워드라인은 예비셀블럭의 테이블(410)의 제2 레지스터(1)의 저장된 정보 '3'에 의해 제4 단위셀블럭의 워드라인 'WL1'이 지정되는 것이다.

<249> 즉, 예비셀블럭(410)의 레지스터에 저장된 정보는 현재 실행명령어에 의해 활성화된 워드라인에 대한 예비워드라인이 어떤 단위셀블럭에 있는지에 관한 정보이다.

<250> 이어서 태그블럭(430)의 셀블럭어드레스 디코더부(431)는 예비셀블럭 테이블(410)에서 출력되며, 현재 실행명령에 의해 활성화되는 워드라인에 대한 예비워드라인이 지정되어 있는 셀블럭어드레스(Extra_BA)를 입력받아서 디코딩하여 단위태그테이블 하나를

선택한다. 여기서 선택된 단위태그테이블에 현재 실행명령어에 의한 논리적 셀블럭어드레스(Cur_LBA)를 현재 실행 명령어에 대응하는 로컬어드레스(Cur_RA)에 따라 선택된 레지스터에 저장시킨다.

<251> 예를 들어 셀블럭어드레스 디코더부에 의해 제2 단위태그테이블(432b)이 선택되고, 현재 실행명령어에 의한 로컬어드레스(Cur_RA)가 '2'이며, 논리적 셀블럭어드레스(Cur_LBA)가 '1'이라면 제2 단위태그테이블(432b)의 제2 레지스터(1)에 2가 저장되는 것이다. (a₀-3구간)

<252> 이어서 a₀-4구간에서 제1 명령어(CD0)에 의해 선택된 단위셀블럭이 이전 명령어에 의해 선택된 단위셀블럭과 동일 단위셀블럭인 지를 비교하고, 그 결과에 의해 예비셀블럭 테이블(410)을 업데이트한다. 이 구간에 대한 설명은 두번째 타이밍의 a₁-4구간에서 자세히 설명한다. (a₀-4구간)

<253> 계속해서, 첫번째 타이밍(t₀)에서 제1 명령어(CD0)에 따라 셀블럭(500)이 동작하는 구간(b₀구간)을 살펴본다.

<254> b₀-1구간에서는 변환된 물리적 셀블럭어드레스(Cur_PBA)에 대응하여 선택된 제1 단위셀블럭에서 로컬어드레스에 대응하는 제1 워드라인을 활성화시킨다. (b₀-1구간)

<255> 이어서, b₀-2구간에서 제1 워드라인에 대응되는 K개의 제1 데이터를 감지증폭한다. (b₀-2구간)

<256> 이어서 b₀-3구간에서 제2 워드라인에 대응하여 감지 증폭된 K개의 데이터를 데이터 래치부(450)에 래치시킨다. (b₀-3구간) 첫번째 타이밍(t₀)에서는 캐패시터를

단위메모리 소자로 사용하는 메모리 장치의 특성상 선택된 워드라인에 대응하는 K개의 데이터를 감지 증폭하는 동작(b_0-2)이 가장 많은 시간이 소요된다.

<257> 한편, 제1 명령어(CD0)가 리드명령어인 경우는 데이터래치부(450)에 래치된 K개의 데이터중 컬럼어드레스에 의해 선택된 데이터를 제2 명령어(CD1)에 대응하는 출력데이터로 출력하고, 제2 명령어(CD1)가 라이트명령어인 경우에는 외부에서 입력되는 데이터로 데이터래치부(450)에 래치된 K개의 데이터중 컬럼어드레스에 의해 선택된 데이터를 덮어쓰게 된다.

<258> 여기서 제1 명령어(CD0)에 대응하는 출력데이터가 실제로 메모리 외부로 출력되는 구간은 두번째 타이밍(t_1)동안이다. 제2 명령어에 대응하는 4K개의 데이터가 감지, 증폭이 완료되어 데이터래치부에 래치되기 전까지는 제1 명령어(CD0)에 대응하는 출력데이터를 출력시킬 수 있는 것이다.

<259> 계속해서 두번째 타이밍(t_1)에서 태그관련 블록이 동작하는 구간(a_1 구간)을 살펴본다. 두번째 타이밍(t_1)에서 다음에 실행될 제2 명령어(CD1)가 입력된다.

<260> a_1-1 구간에서는 제2 명령어(CD1)에 대응하여 입력되는 로우어드레스를 입력되어 로컬어드레스와 논리적 셀블럭어드레스로 분리된다. (a_1-1 구간)

<261> 이어서 a_1-2 구간에서는 태그블럭(430)에서 논리적 셀블럭어드레스(Cur_LBA)를 물리적 셀블럭어드레스(Cur_PBA)로 변환한다. (a_1-2 구간)

<262> 이어서 a_1-3 구간에서는 예비블럭테이블(410)에서 제2 명령어(CD1)에 의해 입력된 로컬어드레스(Cur_RA)에 해당되는 제2 워드라인(예컨대 WL3)의 예비워드라인이 어떤단위 셀블럭에 있는지 감지하여 예비워드라인이 있는 셀블럭어드레스(Extra_BA)를 제어부

(420)와 태그블럭(430)으로 출력한다. 이어서 제2 워드라인(WL3)의 예비워드라인에 의해 선택된 단위태그테이블의 레지스터에 제2 명령어(CD1)에 의해 입력된 논리적 셀블럭어드레스(Cur_LBA)를 업데이트한다.

<263> 예를 들어 제2 명령어(CD1)에 의한 로컬어드레스가 '3'이라면, 제2 워드라인으로 'WL3'이 선택될 것이다. 제2 명령어(CD1)에 의한 논리적 셀블럭어드레스가 '4'이고, 제2 워드라인 'WL3'에 대한 예비워드라인이 제2 단위셀블럭의 'WL3'으로 지정되었다면 이 구간에서 단위태그테이블(432b)의 제4 레지스터(3)에 논리적 셀블럭어드레스 '4'이 저장되는 것이다.(a₁-3구간)

<264> 이어서 a₁-4구간에서 제2 명령어(CD1)에 의해 선택된 단위셀블럭이 이전 명령어인 제1 명령어(CD0)에 의해 선택된 단위셀블럭과 동일 단위셀블럭인 지를 비교한다.

<265> 비교한 결과 제1 명령어(CD0) 및 제2 명령어(CD1)에 의해 연속해서 같은 단위셀블럭을 액세스하게 된다면, 제1 명령어(CD0)에 의해 선택된 제1 워드라인의 예비워드라인을 제1 명령어(CD0)에 의해 입력된 논리적 셀블럭어드레스가 저장되어 있던 단위셀블럭의 제1 워드라인으로 교체한다. 여기서 예비워드라인의 교체는 예비셀 테이블에 저장된 정보를 업데이트하는 것을 말한다. 여기서 업데이트하는 이유는 연속적으로 같은 물리적 셀블럭을 액세스하기 때문에 제1 명령어(CD0)에 대응하는 재저장동작을 제1 워드라인에 대응하여 지정된 예비워드라인에서 수행하기 위함때문이다.

<266> 예를 들어 제2 명령어(CD1)가 입력되는 순간에는 제1 명령어(CD0)에 대응하는 워드라인 'WL1'의 예비워드라인이 제4 단위셀블럭의 워드라인 'WL1'이었는데, 업데이트후에는 워드라인 'WL1'의 예비워드라인이 제1 단위셀블럭의 'WL1'이 되는 것이다. 즉, 예비

셀블럭 테이블(410)의 제2 레지스터(1)에는 '3'이 저장되어 있었는데, 업데이터 이후에는 제2 레지스터(1)에 '0'이 저장되는 것이다.

<267> 계속해서 살펴보면, 제1 명령어(CD0)에 의해 선택된 제1 워드라인에 대응하여 감지, 증폭되어 단위셀블럭의 비트라인 센스앰프와 비트라인에 인가된 K개의 제1 데이터를 강제프리차지시켜 제거한다. 여기서 제1 워드라인에 대응하는 K개의 제1 데이터는 데이터래치부(450)에 래치 되어있기 때문에 단위셀블럭의 비트라인 센스앰프에 저장된 K개의 제1 데이터는 강제 프리차지하여 제거하여도 상관없다. 강제프리차지 하는 이유는 제1 명령어에 의해 액세스된 단위셀블럭에 제2 명령어(CD1)에 의한 제2 워드라인을 바로 활성화시키기 위한 것이다.(a₁-4구간)

<268> 계속해서 두번째 타이밍(t1)에서 셀블럭(500)이 동작하는 구간(b₁구간)을 살펴본다

<269> b₁-1구간에서 제2 명령어(CD1)에 의해 입력된 로컬어드레스에 의해서 선택된 제2 워드라인을 활성화시킨다. 또한 이 때 제1 명령어(CD0)에 의해 a₀-3구간에서 선택된 제1 워드라인에 대한 예비워드라인을 활성화시킨다.(b₁-1구간)

<270> 이어서 b₁-2구간에서 데이터래치부(450)에 래치되어있던 K개의 제1 데이터를 활성화된 제1 워드라인의 예비워드라인에 대응되는 K개의 단위셀에 재저장시킨다. 또한 활성화된 제2 워드라인에 대응하는 K개의 데이터를 감지 증폭한다.(b₁-2구간)

<271> 이어서 b₁-3구간에서 제2 워드라인에 대응하여 감지 증폭된 K개의 데이터를 데이터래치부(450)에 래치시킨다. 제2 명령어(CD1)가 리드명령어인 경우는 데이터래치부(450)에 래치된 K개의 데이터중 컬럼어드레스에 의해 선택된 데이터를 제2 명령어(CD1)에 대

응하는 출력데이터로 출력하고, 제2 명령어(CD1)가 라이트명령어인 경우에는 외부에서 입력되는 데이터로 데이터래치부(450)에 래치된 K개의 데이터중 컬럼어드레스에 의해 선택된 데이터를 덮어쓰게 된다.(b₁-3구간)

<272> 두번째 타이밍(t1)에서도 캐패시터를 단위메모리 소자로 사용하는 메모리 장치의 특성상 제1 명령어(CD0)에 따른 제1 데이터를 재저장하는 시간과 제2 명령어(CD1)에 따른 제1 데이터를 감지, 증폭하는 시간이 대부분을 차지하게 된다.

<273> 본 발명의 메모리 장치는 전술한 바와 같이 한 단위셀블럭에 제1 및 제2 데이터가 연속해서 액세스되면 제1 데이터에 관한 재저장동작을 수행하지 않고 제2 데이터에 관한 액세스 동작을 수행하고, 제1 데이터에 관한 재저장동작은 준비된 예비워드라인에 의해 수행하게 되는 것이다. 이렇게 함으로서 데이터 재저장하는 시간만큼 데이터를 고속으로 액세스할 수 있다.

<274> 계속해서 9개의 단위셀블럭중에서 선택된 제1 및 제2 단위셀블럭에 제1 데이터 및 제2 데이터가 교대로 액세스 되는 경우를 살펴본다.

<275> 먼저 제1 데이터를 상기 9개의 단위셀블럭중에서 선택된 제1 단위셀블럭에서 감지 증폭한 다음 데이터래치부(450)에 래치시킨다. 이어서 제1 단위셀블럭에 제1 데이터를 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 9개의 단위셀블럭중에서 선택된 제2 단위셀블럭에서 제2 데이터를 감지 증폭하여 상기 데이터래치부에 래치시킨다.

<276> 따라서 본 발명의 메모리장치는 데이터가 하나의 셀에 계속해서 액세스되는 경우에는 이전 데이터에 대한 재저장은 지정된 예비워드라인에 대응하는 단위셀에서 수행하고,

데이터가 단위셀에 교대로 액세스되는 경우에는 인터리빙모드로 동작함으로써, 액세스되는 패턴에 상관없이 고속으로 데이터를 액세스 할 수 있게 되는 것이다.

<277> 또한, 도22에 도시된 구간에 대한 메모리 장치의 동작은 명확하게 구분되는 것이 아니라, 실제 메모리 장치에서는 구현하는 경우에는 각 구간의 동작 특성에 따라 동작하는 구간이 서로 시간적으로 겹칠수 있다. 예를 들어 a_1 -4구간에서 태그관련 블록에서는 제1 명령어와 제2 명령어에서 액세스하는 단위셀블록을 비교하고 비교한 결과에 따라 예비셀블록 테이블을 업데이트하는 한편 셀영역에서는 워드라인을 활성화할 수 있는 것이다.

<278> 도23은 태그관련 블록에 대한 동작타이밍의 다른 실시예를 도시하고 있다.

<279> 도23에 도시된 바와 같이, 태그관련 블록에 관한 동작 타이밍을 로컬어드레스를 입력받아 예비워드라인을 먼저 감지해 내고(A), 태그블록(430)에서 논리적 셀블록어드레스를 물리적 셀블록어드레스로 변환하고(B), 이어서 현재 실행중인 명령어에 따라 액세스되는 단위셀블록과 이전 명령어에 따라 액세스되는 단위셀블록이 동일한지 비교(B)하고, 비교한 결과에 따라서 예비셀블록을 업데이트(G)하도록 메모리 장치를 구현할 수 있다. 이 경우에는 도시된 바와 같이 예비셀블록을 업데이트(G)하는 동안 셀블록에서는 선택된 워드라인을 활성화시키는 등의 동작을 수행할 수 있는 것이다.

<280> 도24은 본 발명의 바람직한 실시예에 따라 구현한 메모리 장치를 나타내는 블록구성도이다.

- <281> 본 실시예에 따른 메모리 장치는 다수의 뱅크를 구비하고 있는데, 각각의 뱅크는 같은 구조를 가지고 있기 때문에, 도24에는 하나의 뱅크(1000)만을 도시하였다. 또한 도 24에는 본 발명의 특징적인 내용을 나타내는데 필요한 블럭만을 도시한 것이다.
- <282> 도24를 참조하여 살펴보면, 하나의 뱅크(1000)에는 총 8개의 세그먼트(1100a ~ 1100h)로 구성되고, 세그먼트(1100a ~ 1100h)에서 출력되는 데이터를 감지, 증폭하여 데이터 입,출력버퍼(3000)로 전달하거나, 데이터 입,출력버퍼(3000)에서 출력되는 데이터를 세그먼트(1100a ~ 1100h)로 전달하기 위한 I/O 센스앰프부(1200, 1300)를 구비한다.
- <283> 또한, 세그먼트(1100a)는 제어부(400)에서 출력되는 어드레스를 디코딩하여 셀영역(500)으로 출력하기 위한 로우어드레스 디코더부(800)와, 컬럼어드레스를 입력받아 셀영역(500)으로 출력하기 위한 컬럼어드레스 디코더부(900)를 구비한다.
- <284> 도24에 도시된 메모리 장치는 용량이 256Mb인 경우를 나타내고 있는데, 4개의 뱅크로 구성되는 경우 하나의 뱅크에는 64Mb의 단위셀이 구성되며, 총 8개의 세그먼트가 하나의 뱅크를 구성하고, 하나의 세그먼트는 8Mb로 구성된다.
- <285> 세그먼트마다 구비되는 셀블럭(500)은 9개의 단위셀블럭(510~590)으로 구성되고, 하나의 단위셀블럭은 256개의 워드라인과 4K(4×1024)개의 비트라인을 구비하고, $256 \times 4K$ 개의 단위셀을 구비한다.
- <286> 제어부(400)는 9개의 단위셀블럭(510~590)에 대한 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하기 위한 태그블럭(430)과, 256개의 예비 워드라인 정보를 저장하기 위한 예비셀블럭 테이블(410)과, 태그블럭(430) 및 예비셀블럭 테이블(410)을 제어하기 위한 태그제어부(420)을 구비한다.

- <287> 셀블럭(500)에서는 입력되는 셀블럭어드레스에 대응하는 단위셀블럭보다 하나를 더 구비하기 때문에, 입력되는 셀블럭어드레스는 논리적 셀블럭어드레스로 인식하고, 제어부(400)의 태그블럭(430)에서 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하는 동작이 수행하는 것이다. 또한 제어부(400)에서는 입력되는 로우어드레스에 의해 선택된 하나의 워드라인에 대응하는 예비워드라인이 어떤 단위셀블럭에 지정되었는지를 찾는 동작을 수행한다.
- <288> 또한 셀블럭(500)에 구비된 9개의 단위셀블럭(예를 들어 510)은 각각 로컬비트라인 센스앰프부(510a, 510c)를 구비하는데, 하나의 단위셀블럭은 선택된 워드라인에 대응하는 K개의 데이터중 일부는 일측에 구비된 로컬비트라인 센스앰프부(510a)에서 감지 증폭하고, 나머지 데이터는 타측에 구비된 로컬비트라인 센스앰프부(510c)에서 감지 증폭하는 구조로 구성된다.(도10 참조)
- <289> 또한 셀블럭(400)은 9개의 단위셀블럭(510~590)중 선택된 하나의 단위셀블럭에서 구비된 로컬비트라인 센스앰프부로부터 출력되는 데이터를 래치하기 위한 제1 및 제2 글로벌비트라인 센스앰프부(710,730)와, 제1 및 제2 글로벌비트라인 센스앰프부(710,730)와 9개의 단위셀블럭(510~590)간의 데이터 이동을 위한 글로벌비트라인(720)과, 글로벌비트라인(720)과 9개의 단위셀블럭(510~590)을 연결하기 위한 5개의 글로벌비트라인 연결부(610~650)를 구비하고 있다.(도9 참조)
- <290> 제1 및 제2 글로벌비트라인 센스앰프부(710,730)는 기본적으로 로컬비트라인 센스앰프와 센스앰프를 구비하여, 로컬비트라인 센스앰프부로부터 출력되는 데이터를 래치하고, 글로벌비트라인을 통해 전달됨으로서 감소되는 데이터 신호를 증폭하는 역할을 한다.

- <291> 도25는 서로 다른 단위셀블럭으로 데이터를 교대로 액세스할 경우, 인터리브 모드에서의 동작을 보여주는 파형도이다. 도26은 하나의 단위셀블럭에서 데이터를 연속적으로 리드할 때의 동작을 보여주는 파형도이며, 도27은 하나의 단위셀블럭에 데이터를 연속적으로 라이트할 때의 동작을 보여주는 파형도이다.
- <292> 먼저 도25를 참조하여 본 실시예에 따른 메모리 장치의 인터리브 모드에서의 리드 동작을 살펴본다.
- <293> 하나의 명령어에 대응하여 입력되는 로우어드레스는 뱅크 및 세그먼트와, 한 세그먼트내에서 하나의 단위셀블럭을 지정하기 위한 셀블럭어드레스와, 선택된 셀블럭내에서 하나의 워드라인을 선택하기 위한 로컬어드레스를 포함한다. 또한 한 단위셀블럭에 구비된 4K개의 비트라인중 하나를 선택하기 위한 컬럼어드레스도 명령어에 대응하여 입력된다.
- <294> 첫번째 타이밍(t0)에서, 제1 리드명령어(RD0)에 대응하는 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하고, 변환된 물리적 셀블럭어드레스에 의해 제1 단위셀블럭(BL0)이 선택된다.
- <295> 이어서 제1 리드명령어(RD0)에 대응하는 로컬어드레스에 의해 제1 단위셀블럭(BL0)의 워드라인(WL0)이 활성화된다. 활성화된 워드라인(WL0)에 대응하는 4K개의 단위셀에 저장된 4K개의 데이터가 제1 단위셀블럭(BL0)에 구비된 로컬비트라인 센스앰프부(510a,510c)에 의해 감지, 증폭된다.
- <296> 감지 증폭된 4K개의 데이터는 글로벌비트라인(720)을 통해 제1 및 제2 글로벌비트라인 센스앰프부(710,730)로 이동되어 래치된다. 이어서, 제1 및 제2 글로벌비트라인 센

스앰프부(710,730)에 래치된 4K개의 데이터중에서, 제1 리드명령어(RD0)에 대응하는 컬럼어드레스에 의해 선택된 하나의 데이터(D0)가 I/O센스앰프부(1200)로 출력된다. 여기서 실제 데이터(D0)가 메모리 장치 외부로 출력되는 타이밍은 두번째 타이밍(t1)이다.

<297> 이어서 두번째 타이밍(t1)에서 제1 리드명령어(RD0)에 대응하여 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치된 4K개의 데이터는 제1 단위셀블럭(BL0)의 워드라인(WL0)에 대응하는 4K개의 단위셀로 이동되어 재저장된다.

<298> 한편, 제1 리드명령어(RD0)에 대응하여 데이터의 재저장 동작이 수행되고 있는 동안에, 제2 리드명령어(RD1)에 대응하여 선택된 제2 단위셀블럭(BL1)의 워드라인(WL0)이 활성화되고, 활성화된 워드라인(WL0)에 대응하는 4K개의 데이터가 제2 단위셀블럭(BL1)에 구비된 로컬비트라인 센스앰프부(520a,520c)에 의해 감지, 증폭된다.

<299> 이어서, 제2 리드명령어(RD1)에 대응하여 감지, 증폭된 4K개의 데이터는 글로벌비트라인(720)을 통해 제1 및 제2 글로벌비트라인 센스앰프부(710,730)로 이동되어 래치된다.

<300> 이어서, 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치된 4K개의 데이터중에서, 제2 리드명령어(RD1)에 대응하는 컬럼어드레스에 의해 선택된 하나의 데이터(D1)가 I/O센스앰프부(1200)로 출력된다. 여기서 실제 데이터(D1)가 메모리 장치 외부로 출력되는 타이밍은 세번째 타이밍(t2)이다.

<301> 이어서 세번째 타이밍(t2)에서 제2 리드명령어(RD0)에 대응하여 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치된 4K개의 데이터는 제2 단위셀블럭(BL1)의 워드라인(WL0)에 대응하는 4K개의 단위셀로 이동되어 재저장된다.

- <302> 따라서 본 실시예에 따른 메모리 장치는 제1 단위셀블럭(BL0)과 제2 기본셀블럭(BL1)에 데이터가 교대로 액세스될 때에는 하나의 단위셀블럭에서 데이터를 재저장하는 동안 다른 단위셀블럭에서는 감지, 증폭동작을 수행하는 것이다.
- <303> 이와 같이 단위셀블럭간 교대로 데이터가 리드될 때에는 인터리빙 모드로 동작함으로써 데이터 재저장시간만큼 데이터 리드시간을 줄일 수 있어 고속으로 데이터 액세스가 가능해진다. 도25에 도시된 단위셀블럭간 인터리빙에 의한 로우사이클 타임은 한번의 명령어 입력으로 워드라인이 활성화되고 난후, 다음명령어에 대응하는 워드라인이 활성화되는 시간을 말한다.
- <304> 또한 단위셀블럭간에 교대로 데이터가 라이트되는 동작은, 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치된 4K개의 데이터중에서 라이트명령어에 대응하여 선택된 데이터를 라이트명령어에 대응하여 입력된 입력데이터로 덮어쓰는 동작을 제외하고는, 리드때의 동작과 같기 때문에 생략한다.
- <305> 이어서 도26을 참조하여 하나의 단위셀블럭에 연속적으로 데이터를 리드할 때의 동작을 살펴본다.
- <306> 도26에 도시된 바와 같이 총 8번의 리드명령어(RD0~RD7)가 입력되는 것으로 가정하고, 각각의 리드명령어에 표시된 괄호안의 내용은 데이터가 리드될 단위셀블럭의 워드라인을 나타낸다. 8번의 리드명령어(RD0~RD7)에 대응하는 워드라인의 예비워드라인은 제2 단위셀블럭(BL1)에 있는 것을 가정한다.
- <307> 첫번째 타이밍(t0)에서, 입력된 제1 리드명령어(RD0)에 대응하는 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환한다. 각 타이밍에서 빗금친부분이 논리적 셀블

터어드레스를 물리적 셀블터어드레스로 변환하는 구간을 나타낸다. 여기서는 어드레스 변환된 물리적 셀블터어드레스에 의해 선택되는 단위셀블터는 논리적 셀블터어드레스에 의해 선택되는 단위셀블터와 같은 것으로 가정한다.

<308> 제1 리드명령어(RD0)에 대응하여 변환된 물리적 셀블터어드레스로 인해 제1 단위셀블터(BL0)의 워드라인(WL0)이 활성화 된다. 이어서 활성화된 워드라인(WL0)에 대응하는 4K개의 단위셀에 저장된 데이터가 제1 단위셀블터(BL0)에 구비된 로컬 센스앰프부에 의해 감지 증폭된다.

<309> 이어서 로컬 센스앰프부에 의해 감지 증폭된 4K개의 데이터는 글로벌비트라인(720)을 통해 제1 및 제2 글로벌비트라인 센스앰프부(710,730)으로 이동되어 래치된다. 제1 및 제2 글로벌 센스앰프부(710,730)에 래치된 4K개의 데이터중에서 제1 리드명령어(CD0)에 대응하여 입력된 컬럼어드레스에 의해 선택된 하나의 데이터(D0)가 I/O 센스앰프(1200)로 출력된다. 이 때 I/O센스앰프부(1200), 데이터 입출력버퍼(3000)등을 통해 메모리 장치의 외부로 데이터(D0)가 실제 출력되는 구간은 두번째 타이밍(t1)이다.

<310> 또한 첫번째 타이밍(t0)에서 워드라인(WL0)의 예비워드라인이 제2 단위셀블터(BL1)의 워드라인(WL0)임을 감지하게 된다.

<311> 이어서 두번째 타이밍(t1)에서 제1 단위셀블터(BL0)에 구비된 로컬비트라인 센스앰프부와 비트라인을 강제프리차지시켜, 제1 리드명령어(RD0)에 대응하여 감지 증폭되어 제1 단위셀블터(BL0)의 로컬비트라인 센스앰프부에 저장된 4K개의 데이터를 제거한다.

<312> 이어서 제2 리드명령어(RD1)에 대응하는 어드레스에 따라서 제1 단위셀블터(BL0)의 워드라인(WL1)을 활성화시킨다. 이어서 활성화된 워드라인(WL1)

에 대응하는 4K개의 단위셀에 저장된 데이터가 제1 단위셀블럭(BL0)에 구비된 로컬비트 라인 센스앰프부에 의해 감지 증폭된다.

<313> 한편, 이 타이밍(t1)에서 제1 및 제2 글로벌센스앰프부(710,730)에 래치된 4K개의 데이터 -제1 리드명령어(RD0)에 의해 재저장될 4K개의 데이터- 는 워드라인(WL0)의 예비 워드라인으로 지정된 제2 기본셀블럭(BL1)의 워드라인(WL0)에 대응되는 4K개의 단위셀로 이동하여 재저장된다. 데이터이동은 글로벌비트라인(720)을 통해 이동되며, 도15에 도시된 'intw0'가 상기의 과정을 나타낸다.

<314> 즉, 제2 리드명령어(RD1)에 대응하는 4K개의 데이터가 제1 단위셀블럭(BL0)에서 감지증폭되는 동안 제1 리드명령어(RD0)에 대응하여 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치된 4K개의 데이터는 제2 단위셀블럭(BL1)의 로컬비트라인 센스앰프부로 이동되고, 워드라인(WL0)에 대응하는 4K개의 단위셀에 재저장되는 것이다.

<315> 또한 이 때부터 제2 단위셀블럭(BL1)의 워드라인(WL0)이 논리적인 제1 단위셀블럭(BL0)이 되고, 제1 기본셀블럭(BL0)의 워드라인(WL0)은 워드라인(WL0)의 예비워드라인으로 지정된다. 여기에 관한 정보가 예비셀블럭 테이블(410)과 태그블럭(430)에 각각 업데이트되며, 도26의 도면부호 'EBT_UP1'가 이 때의 업데이트 동작을 나타내는 것이다.

<316> 이어서 제2 리드명령어(RD1)에 대응하여 감지 증폭된 4K개의 데이터가 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 이동되어 래치되고, 래치된 데이터 중에서 제2 리드명령어(RD1)에 대응하여 입력된 컬럼어드레스에 의해 선택된 데이터(D1)가 I/O 센스앰프(1200)로 출력된다. 이 때에도 I/O센스앰프, 출력버퍼등을 통해 메모리 장치의 외부로 데이터(D1)가 실제 출력되는 구간은 세번째 타이밍(t2)이다.

- <317> 이어서 세번째 타이밍(t2)에서 제3 리드명령어(RD2)에 대응하는 제1 단위셀블럭(BL0)에 대응하는 워드라인(WL2)을 활성화시키고, 활성화된 워드라인(WL2)에 대응하는 4K개의 데이터를 감지 증폭한 다음, 제1 및 제 2 글로벌비트라인 센스앰프부(710,730)에 이동되어 래치한다.
- <318> 한편, 제2 리드명령어(RD1)에 대한 재저장동작이 제2 단위셀블럭(BL1)의 워드라인(WL1)에 대응하는 4K개의 단위셀에서 수행되며, 도26에 도시된 'intw1'가 상기의 과정을 나타낸다. 이 때에도 예비셀블럭 테이블(410) 및 태그테이블(430)를 업데이트하게 되며 도26의 'EBT_UP2'가 이 때의 업데이트과정을 나타낸다.
- <319> 네번째 타이밍(t3)에서 제3 리드명령어(RD2)에 대한 재저장동작을 제2 단위셀블럭(BL1)의 워드라인(WL0)에 대응하는 4K개의 단위셀에서 수행되며, 'intw2'가 상기의 과정을 나타낸다. 이 때에도 각각의 타이밍에서 예비셀블럭테이블(430) 및 태그테이블(430)을 업데이트 하게 되며, 'EBT_UP3'가 이 때의 업데이터 과정을 나타낸다.
- <320> 한편으로 제4 리드명령어(RD3)에 대응하는 제1 단위셀블럭(BL0)의 워드라인(WL3)을 활성화시키고 활성화된 워드라인(WL3)에 대응하는 4K개의 데이터를 감지 증폭한다.
- <321> 이어서 다섯번째 타이밍(t4)에서 제4 리드명령어(RD3)에 대한 재저장동작은 데이터를 워드라인(WL3)의 예비워드라인이 있는 제2 단위셀블럭(BL1)으로 이동시키지 않고, 제1 단위셀블럭(BL0)의 워드라인(WL3)에 대응하는 4K개의 단위셀에서 수행한다.
- <322> 이는 다음 명령어인 제5 리드명령어(RD4)에 의해 동작되는 단위셀블럭이 제2 단위셀블럭(BL1)이고, 이전 명령어인 제4 리드명령어(RD3)에 의해 동작되고 있는 단위셀블럭이 제1 단위셀블럭(BL0)이기 때문에, 다섯번째 타이밍(t4)에서 제5 리드명령어(RD4)에

대응되는 워드라인(WL0)을 바로 활성화시킬 수 있기 때문이다. 따라서 이 때에는 워드라인(WL3)의 예비워드라인이 변경되지 않기 때문에 예비셀블럭테이블(430)을 업데이트할 필요가 없고, 이를 나타내는 것이 'EBT_UPN'이다.

<323> 계속해서 살펴보면, 제5 및 제6 리드명령어(RD4~RD5)가 각 타이밍(t_4, t_5)에 입력되면, 전술한 바와 같이 첫타이밍에서 리드명령어에 대응하는 4K개의 데이터를 감지 증폭하여, 글로벌비트라인 센스앰프에 래치시키고, 래치된 4K의 데이터 중에서 선택된 한 데이터를 출력시킨다. 이어서 두번째 타이밍에서는 활성화된 워드라인의 예비워드라인이 지정된 단위셀블럭으로 데이터를 이동시켜 재저장동작을 수행한다.

<324> 이어서 제7 및 제8 리드명령어(RD6, RD7)가 타이밍(t_6, t_7)에 각각 입력되면 인터리브모드로 동작하게 된다.

<325> 명령어실행 구간을 나타내는 부분에서 점선부분은 실제 입력된 리드명령어에 따른 동작이 수행되고 있지만, 내부적으로는 해당되는 데이터는 예비워드라인이 지정된 단위셀블럭으로 이동되어 재저장되며, 그에 따라서 예비블럭테이블 (410) 및 태그테이블(430)이 업데이트되는 구간이다. 'EBT_UP1 ~ EBT_UP5'는 태그블럭이 업데이트되는 것을 나타내는 신호이고, 'EBT_UPN1 ~ EBT_UPN2'는 예비태그테이블이 업데이트되지 않는다는 것을 나타내는 신호이다.

<326> 또한 빗금친 부분을 나타내는 'X' 구간은 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하는 동작과 함께, 동일 단위셀블럭에 연속해서 데이터를 리드함으로써 이전 명령어에 따라 감지 증폭되어 단위셀블럭의 로컬비트라인 센스앰프부에 래치된 데이터를 강제프리차지하는 시간을 나타내는 것이다.

- <327> 결론적으로, 본 실시예에 따른 메모리 장치는 같은 단위셀블럭에 연속적으로 데이터를 리드할 경우에, 첫번째 타이밍에서 입력된 리드명령어에 대응하는 데이터를 증폭시킨 다음 글로벌비트라인 센스앰프부로 이동시켜 래치시키고, 두번째 타이밍에서 리드명령어에 의해 활성화된 워드라인의 예비워드라인이 지정된 단위셀블럭으로 글로벌비트라인 센스앰프부에 래치되어 있던 데이터를 옮겨서 재저장하게 된다.
- <328> 따라서 연속해서 같은 단위셀블럭에 데이터를 리드하는 경우에도 데이터 재저장 동작에 관계없이 다음 실행되는 리드명령어를 입력받을 수 있어 고속으로 데이터를 리드시킬수 있는 것이다.
- <329> 이어서 도27를 참조하여 하나의 단위셀블럭에서 데이터를 연속적으로 라이트할 때의 동작을 살펴본다.
- <330> 첫번째 타이밍(t0)에서 제1 라이트명령어(WR0)에 대응하여 입력된 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환시켜, 제2 단위셀블럭(BL1)의 워드라인(WL0)을 활성화시킨다.
- <331> 여기서 제1 라이트명령어(WR0)에 대응하여 입력된 셀블럭어드레스는 논리적 제1 단위셀블럭(BL0)의 워드라인(WL0)을 선택하는 것인데, 논리적 제1 단위셀블럭(BL0)의 워드라인(WL0)에 대응하는 실제 데이터는 물리적인 제2 단위셀블럭(BL1)의 워드라인(WL0)에 대응하는 단위셀에 저장되어 있다는 것을 나타낸다. 도27에 도시된 '명령어실행 구간'에는 라이트명령어에 대응하여 입력되는 셀블럭어드레스의 변환동작에 의해 입력되는 논리적인 단위셀블럭의 워드라인에 관한 정보와 변환뒤에 선택되는 물리적인 단위셀블럭의 워드라인이 표시되어 있다.

- <332> 이어서 활성화되는 워드라인(WL0)에 대응되는 4K의 데이터를 감지 증폭하고, 제1 및 제2 글로벌비트라인 센스앰프부(710,730)로 이동시키는 래치시킨다.
- <333> 이어서 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치된 4K개의 데이터중에서 제1 라이트명령어(WR0)에 대응하여 입력된 컬럼어드레스에 의해 선택된 하나의 데이터를 제거하고, 외부에서 제1 라이트명령어(WR0)에 대응하여 입력된 입력데이터(D0)를 제거된 데이터가 대신에 래치시킨다.
- <334> 두번째 타이밍(t1)에서 제2 단위셀블럭(BL0)의 워드라인(WL0)에 대응하는 4K개의 단위셀에 제1 라이트명령어(WR0)에 의해 감지 증폭된 4K개의 데이터를 재저장시킨다. 따라서 데이터를 감지 증폭하고 래치한 뒤에, 래치된 데이터중 하나를 외부에서 입력된 데이터로 교환하는 동작만 제외하고는 리드명령어 동작과 같은 방식으로 라이트동작이 진행되는 것이다.
- <335> 한편 이 타이밍(t1)에 제2 라이트명령어(WR1)에 대응하여 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환시키고, 변환된 물리적 셀블럭어드레스에 따라 제3 단위셀블럭(BL2)의 워드라인(WL1)을 활성화시킨다. 이어서 활성화된 워드라인(WL1)에 대응하는 4K개의 데이터를 감지 증폭하고 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치시킨다.
- <336> 이어서 제2 라이트명령어(WR1)에 대응하여 입력된 컬럼어드레스에 의해 선택된 하나의 데이터를 제거하고, 외부에서 제2 라이트명령어(WR1)에 대응하여 입력된 입력데이터(D1)를 제거된 데이터 대신에 래치시킨다.

- <337> 세번째 타이밍(t2)에서 제2 라이트명령어(WR1)에 대응하여 제1 및 제2 글로벌비트 라인 센스앰프부(710,730)에 래치된 4K개의 데이터를 제3 단위셀블럭(BL2)의 워드라인(WL1)에 해당되는 4K개의 단위셀에 재저장시킨다.
- <338> 다른 한편으로 제3 라이트명령어(WR2)에 대응하여 제2 단위셀블럭(BL1)의 워드라인(WL2)을 활성화시킨다. 이어서, 활성화된 워드라인(WL2)에 대응하는 4K개의 데이터를 감지 증폭하고 제1 및 제2 글로벌비트라인 센스앰프부(710,730)에 래치시킨다.
- <339> 이어서 제3 라이트명령어(WR2)에 대응하여 입력된 컬럼어드레스에 의해 선택된 하나의 데이터를 제거하고, 외부에서 제3 라이트명령어(WR2)에 대응하여 입력된 입력데이터(D0)를 제거된 데이터 대신에 래치시킨다.
- <340> 네번째 타이밍(t3)에서 제3 라이트명령어(WR2)에 대응하여 제1 및 제2 글로벌비트 라인 센스앰프부(710,730)에 래치된 4K개의 데이터는 제2 단위셀블럭(BL1)에 저장되지 않고, 워드라인(WL2)의 예비워드라인으로 지정된 제1 단위셀블럭(BL0)의 워드라인(WL2)에 대응되는 4K개의 단위셀에 재저장된다. 도27에 도시된 'intw0'가 상기의 과정을 나타낸다.
- <341> 이 때 워드라인(WL2)의 예비워드라인이 제1 단위셀블럭(BL0)의 워드라인(WL2)에서 제2 단위셀블럭(BL1)의 워드라인(WL2)로 바뀌게 되며, 이에 관한 정보가 예비셀블럭 테이블(410)에 업데이트 된다. 또한 논리적 셀블럭어드레스와 관한 정보도 바뀌었으므로 태그테이블(430)도 업데이트 되며, 이에 관한 것을 나타내는 것이 'EBT_UP1'이다.
- <342> 이제부터는 워드라인(WL2)의 예비워드라인은 제2 단위셀블럭(BL0)의 워드라인(WL2)이 된다. 또한, 이전까지 제2 단위셀블럭(BL1)의 워드라인(WL2)이 논리적 셀블럭어드레

스 'BL0'의 워드라인(WL2)이었어나, 이제부터는 제1 단위셀블럭(BL0)의 워드라인(WL2)이 논리적 셀블럭어드레스 'BL0'의 워드라인(WL2)을 나타내게 된다.

- <343> 이후에 제4 라이트명령어(WR3)에 대한 동작은 전술한 제1 라이트명령어(WR0)에 대한 동작과 동일하게 이루어진다.
- <344> 이어서, 제5 라이트명령어(WR4)와, 제6 라이트명령어(WR5)가 수행될 때에도 연속해서 제1 단위셀블럭(BL0)을 데이터를 라이트해야 되는 상황이기 때문에, 상기와 같은 동작으로 여섯번째 타이밍(t5)에서 제5 라이트명령어(WR4)에 대응하여 래치된 4K개의 데이터가 워드라인(WL0)의 예비워드라인이 지정된 제2 단위셀블럭(BL1)의 워드라인(WL0)에 대응되는 4K개의 단위셀에 재저장된다. 도27에 도시된 'intw1'가 상기의 과정을 나타낸다.
- <345> 도27에 도시된 명령어실행 구간의 점선부분은 다음에 실행될 라이트명령어가 같은 단위셀블럭에 데이터를 라이트하기 때문에 이전 명령어에 대응하는 데이터가 예비워드라인이 지정된 단위셀블럭으로 이동되어 재저장 동작이 일어나는 구간을 나타낸다.
- <346> 결론적으로 본 실시예에 따른 메모리 장치는 서로 다른 단위셀블럭으로 데이터를 라이트할 경우에는 인터리빙 방법을 사용하여 데이터를 라이트하고, 연속해서 같은 단위셀블럭으로 데이터를 라이트하는 경우에는 예비워드라인이 지정된 단위셀블럭으로 데이터가 이동되어 재저장 동작이 수행되는 것이다. 이는 입력되는 셀블럭어드레스에 대응되는 단위셀블럭보다 하나의 단위셀블럭을 더 추가하여 256개의 예비워드라인을 구비하고 있기 때문에 가능한 것이다.

- <347> 따라서 데이터를 라이트하는 동작에서 데이터 재저장하는 시간에 상관없이 다음 라이트명령어를 수행할 수 있으므로 고속으로 데이터를 라이트 할 수 있다.
- <348> 또한, 하나의 단위셀블럭에 계속해서 데이터가 라이트되던지 서로 다른 단위셀블럭에 데이터가 교대로 라이트되는지, 데이터가 라이트 되는 패턴에 상관없이 항상 데이터 재저장되는 시간만큼이 줄어든게 되어, 고속 데이터 라이트가 가능한 것이다.
- <349> 도28은 도25의 메모리 장치에서 도26에 도시된 순서대로 리드명령어(RD0~RD7)가 입력될 때의 시뮬레이션 파형도이다.
- <350> 도28을 참조하여 살펴보면, 리드명령어(RD0~RD2, RD4~RD5)가 실행될 때에는 다음 리드명령어가 계속 같은 셀블럭을 리드하기 때문에, 리드된 데이터의 재저장동작은 예비위드라인이 지정된 단위셀블럭에서 수행되며, 총 5번의 데이터 이동 및 재저장동작(도28의 'intW')이 일어난다. 또한 리드명령어(RD3, RD6, RD7)가 실행될 때에는 다음 리드명령어가 다른 단위셀블럭을 액세스하기 때문에, 선택된 하나의 단위셀블럭에서 리드된 데이터의 재저장동작(도28의 'act')이 일어나는 것을 알 수 있다.
- <351> 도29는 도24에 도시된 글로벌비트라인 연결부(610 ~ 650)를 제어하기 위한 신호(예컨대 GBIS12)를 생성하기 위한 회로도의 일예이다.
- <352> 도29에 도시된 부분은 글로벌비트라인 연결부(610 ~ 650, 도10의 73_1~73_3 참조)를 제어하기 위한 회로로서, 제1 노어게이트(NOR1)에서 명령어에 의해 선택된 위드라인에 대응하는 데이터의 감지 증폭을 위한 신호(pcg-Act[2n], pcg-Act[2n+1])를 입력받고, 제2 노어게이트(NOR2)에서 연속해서 같은 단위셀블럭을 액세스할 경우 이동하여 재저장하라는 신호(intW[2n], intW[2n+1])를 입력받아 글로벌비트라인 연결부를 제어하는 신호

(GBIS)를 출력하게 된다. $\tau 1$, $\tau 2$, $\tau 3$ 은 각각 지연소자(Delay1 ~ Delay3)에서 지연되는 시간이다.

<353> 도30은 도29에 도시된 회로도에서 생성되는 신호에 따라 본 발명의 메모리 장치가 동작하는 것을 보여주는 파형도이다.

<354> 도30에 도시된 빗금친 부분은 각 타이밍별로 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하는 시간을 나타낸다. 도30에 도시된 지연시간($\tau 1$, $\tau 2$, $\tau 3$)은 도29에 도시된 딜레이(Delay1~Delay3)에 따라 정해진다.

<355> 도30에서 'LSA_EN'은 로컬비트라인 센스앰프의 인에이블신호를 나타내는 것이고, 'BIS'는 로컬비트라인 연결부의 인에이블신호이고, 'GBIS'는 글로벌비트라인 센스앰프의 인에이블 신호이고, 'GSA_EN'은 글로벌비트라인 센스앰프의 인에이블 신호이다.

<356> 또한, ' Δ '는 제1 명령어에 의해 활성화된 워드라인을 다음에 실행되는 제2 명령어가 같은 단위셀블럭을 액세스하는 경우 강제로 비활성화시키고, 제1 명령어에 의해 감지 증폭된 데이터를 강제 프리차시시키는 것을 나타내는 것이고, 'intW'는 제1 명령어에 대응하는 워드라인의 예비워드라인이 저장된 단위셀블럭으로 데이터를 이동시켜 재저장하는 것을 나타내는 것이다. 도30에 도시된 바와 같이 본 발명의 메모리 장치에서 셀블럭어드레스 변환시간과 강제 프리차지시간은 전체 한 타이밍에서 약 40% 정도를 차지하는 것을 알 수 있다.

<357> 앞에서 살펴본 바와 같이, 본 발명의 사상을 구현하려고 하면 종래의 메모리 장치 구조에서 하나의 셀블럭당 하나의 단위셀블럭을 추가하고, 예비셀블럭 테이블과 태그테

이블등을 추가로 구비해야 한다. 또한 글로벌비트라인 센스앰프와 글로벌비트라인을 추가로 구비해야 하기 때문에 통상적인 메모리 장치보다 추가적인 면적이 필요하다.

<358> 그러나 상기의 블럭을 추가함으로써 메모리 장치의 동작속도는 재저장하는 시간만큼 감소되어, 종래의 메모리 장치보다 고속으로 데이터를 액세스할 수 있다.

<359> 종래 기술에 의해 메모리 장치의 로우사이클 타임은 t_{RC} 라고 생각했을 때, 본 발명에 따른 메모리 장치의 로우사이클타임은 $\text{MAX}\{0.5*(t_{BAT}+t_{RP}+t_{RC}), t_{INTW}\}$ 로 정해진다. 즉, ' $0.5*(t_{BAT}+t_{RP}+t_{RC})$ '와 ' t_{INTW} '중에서 더 긴 시간을 요하는 것으로 로우사이클타임이 정해진다.

<360> 여기서 t_{BAT} 는 셀블럭어드레스 변환시간이고, ' t_{RP} '는 프리차지 타임이며, ' t_{RC} '는 종래의 로우사이클 타임이고, ' t_{INTW} '는 내부적으로 래치된 데이터가 예비워드라인이 지정된 단위셀블럭으로 이동되어 재저장하는 시간을 나타낸다.

<361> 여기서 ' t_{RP} '는 본발명의 사상을 구현하기 위해 하나의 명령어가 수행될 때마다 하는 강제프리차지 동작으로, 연속해서 같은 셀블럭을 액세스할 때 다음 데이터를 바로 액세스하기 위해 이전에 감지 증폭되어 로컬비트라인 센스앰프에 래치된 데이터를 제거하기 위해 로컬비트라인 센스앰프를 디스에이블시키고, 로컬비트라인을 프리차지시키는 것을 말한다.

<362> 본 발명에 따른 메모리 장치의 로우사이클 타임은 ' $0.5*(t_{BAT}+t_{RP}+t_{RC})$ '와 ' t_{INTW} ' 중에서 소요되는 시간이 더 긴 타이밍으로 정해지는데, 통상적으로 ' $0.5*(t_{BAT}+t_{RP}+t_{RC})$ ' 시간이 t_{INTW} 시간보다는 더 많은 시간이 걸려서, 로우사이클 타임은 ' $0.5*(t_{BAT}+t_{RP}+t_{RC})$ '로 정해진다.

<363> 메모리 장치의 로우사이클 타임이 ' $0.5 \cdot (t_{BAT} + t_{RP} + t_{RC})$ ' 라면, 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하는 시간과 강제프리차지하는 시간이 추가되더라도 데이터 재저장하는 만큼 감소되어 대략 30~40%의 로우사이클 타이밍이 절감되는 효과를 기대할 수 있다.(도26 및 도27에 도시된 본 발명과 종래기술에 의한 로우사이클 타임 참조)

<364> 예를 들어 명령어하나가 수행되는 타이밍이 15n라고 생각하면, 종래에는 두번의 타이밍 즉 30n가 한번 로우사이클타임이었다. 그러나 본 발명에서는 셀블럭어드레스 변환 시간으로 3n, 강제프리차지 시간으로 3n가 추가로 걸린다고 하더라도 한번의 타이밍 15n에 6n가 더해진 21n가 로우사이클 타임이 되는 것이다.

<365> 한편, 전술한 실시예에 의해 메모리 장치는 다음 명령어 수행시 액세스될 단위셀블럭이 이전의 명령어에 의해 액세스된 단위셀블럭과 같을 때만 데이터를 이동시키고, 데이터가 이동시에만 태그를 업데이트 해주는 방법을 사용하였다.

<366> 그런데, 본 발명의 사상을 구현하는데 있어서, 현재 액세스될 단위셀블럭의 워드라인을 활성화시키는 동시에 선택된 워드라인의 예비워드라인이 구비된 단위셀블럭으로 데이터를 항상 이동시켜 재저장시키는 방법을 사용할 수 있다. 이 경우에는 항상 두군데의 단위셀블럭에 같은 데이터가 저장되며, 실제의 데이터가 저장된 셀블럭은 태그테이블에 의해 정해진다. 이 경우에는 매번 명령어를 실행할 때마다 단위셀블럭에서 감지 증폭된 데이터가 글로벌비트라인 센스앰프부에 래치되면, 단위셀블럭의 로컬비트라인 센스앰프부에 래치된 데이터는 강제프리차지하도록 구현하는 것이다.

- <367> 또한, 전술한 실시예에서는 명령어가 실행되는 첫타이밍인 데이터 감지 증폭후 래치시에 데이터를 외부로 출력하거나 데이터를 입력받도록 하였으나, 명령어가 실행되는 두번째 타이밍인 데이터 재저장 동작시에 명령어에 대응하여 데이터를 외부로 출력하거나 입력받도록 메모리 장치를 구현할 수 있다.
- <368> 상기의 방법은 메모리 장치의 설계하는 방법에 따라 적절하게 선택될 수 있으며, 어느 방법으로 메모리 장치를 구현하던지 데이터 재저장하는 시간에 구애받지 않고 데이터를 고속으로 액세스할 수 있게 된다.
- <369> 전술한 실시예에 의한 메모리 장치는 데이터 재저장 시간만큼 데이터 액세스시간이 단축되어 고속 동작이 가능하나, 종래의 메모리 장치에서는 없던 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하는 시간(t_{BAT}) 및 강제프리차지 시간(t_{RP})이 로우사이클 타임에 추가되었다.
- <370> 본 발명에서는 보다 고속으로 데이터를 액세스하기 위해서 도24에 도시된 메모리 장치에서 강제프리차지 시간(t_{RP})과 셀블럭어드레스를 변환하는 시간(t_{BAT})만큼이 줄어든 로우사이클타임을 가지는 메모리 장치를 제안한다.
- <371> 도31은 도23의 메모리 장치에서 강제프리차지 시간(t_{RP})과 셀블럭어드레스를 변환하는 시간(t_{BAT})만큼이 줄어든 로우사이클타임을 가지는 메모리 장치를 나타내는 블럭구성도이다.
- <372> 도31를 참조하여 살펴보면, 본 발명의 메모리 장치는 제어부(400)에서 명령어(CD)를 직접 입력받는 한편, 명령어(CD)를 셀블럭어드레스 변환시간(t_{BAT}) 및 강제프리차지

시간(tRP)만큼 지연시킨, 지연부(460)에서 출력되는 지연된 제2 명령어(CD_D)를 생성한다. 제어부(400)는 명령어(CD)로 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하는 동작과 강제프리차지 동작을 수행하고, 지연된 명령어(CD_D)으로는 데이터 액세스에 필요한 나머지 동작을 수행하도록 셀블럭(500)을 제어한다.

<373> 도32는 도31에 도시된 메모리 장치를 보다 구체적으로 나타낸 블럭구성도이다.

<374> 도32를 참조하여 살펴보면, 본 발명의 메모리 장치는 각각 256개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 8개의 단위셀블럭에, 추가적으로 256개의 워드라인을 더 가지기 위해 추가의 단위셀블럭을 더 포함하여 8+1개의 단위셀블럭으로 구성된 셀블럭(500)과, 9개의 단위셀블럭중에서 선택된 제1 단위셀블럭으로부터 액세스되는 데이터가 제1 단위셀블럭 또는 제1 단위셀 이외의 다른 단위셀블럭으로 재저장되도록 제어하는 데이터엑세스 제어부(460)와, 현재 실행되는 제1 명령어에 대응하는 데이터 액세스 중에, 다음 실행될 제2 명령어에 대응하는 셀블럭어드레스 변환동작과 제1 명령어에 대응하는 비트라인에 대한 강제프리차지 동작이 이루어지도록 데이터엑세스 제어부(460)를 제어하는 명령어 제어부(470)을 구비한다.

<375> 명령어 제어부(470)는 상기 제1 명령어에 대응하여 입력된 로우어드레스에서 논리적 셀블럭어드레스를 감지하여, 9개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하는 셀블럭어드레스 변환부(471)와, 제1 명령어에 의해 감지, 증폭되어 단위셀블럭에 구비된 로컬비트라인센스앰프부에 래치된 데이터를 강제프리차지시키기 위한 강제프리차지 제어부(472)와, 제1 명령어에 대응하여 감지증폭된 데이터가 데이터 래치부(450)에 의해 래치된 이후에 제1 명령어에 대응하는 강제프리차지 동작이 수행되도록 강제프리차지 제어부(472)를 제어하는 명령어 타이밍제어부(473)를 구비한다.

- <376> 명령어타이밍 제어부(473)는 제1 명령어에 대응하는 강제프리차지 동작 및 제2 명령어에 대응하는 셀블럭어드레스 변환동작을 수행하는 시간($t_{RP} + t_{BAT}$)만큼 제2 명령어를 지연시켜 출력하기 위한 지연부(440)을 구비하고, 제2 명령어에 대응하는 데이터의 감지, 증폭동작은 지연부(440)에 의해 지연된 제2 명령어에 의해 수행되도록 데이터억세스 제어부(460)를 제어한다.
- <377> 도33은 도32에 도시된 메모리 장치가 연속적으로 리드명령어를 수행하는 것을 타나내는 동작 파형도이다. 이하에서 도31 내지 도33을 참조하여 메모리장치의 동작을 살펴본다.
- <378> 먼저, 입력된 제1 리드명령어(RD0)에 대응하는 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환하고, 강제프리차지 동작을 수행한다. 이 때 강제프리차지동작은 이전 명령어에 대응하여 액세스되는 단위셀블럭과 현재 실행중인 제1 리드명령어(RD0)에 대해 액세스될 단위셀블럭이 동일하면, 이전 리드명령어에 대한 재저장동작을 수행하지 않고 강제로 프리차지시키는 동작을 말한다. 하지만 여기서와 같이 메모리 장치의 첫번째 리드동작에서는 이전 명령이 없기 때문에 실제로는 강제프리차지 동작은 일어나지 않는다.
- <379> 따라서 본 발명의 메모리장치는 처음 명령어를 수행할 때에 셀블럭어드레스 변환시간(t_{BAT}) 및 강제프리차지시간(t_{RP}) 만큼의 레이턴시(latency)(도33의 'X' 구간)를 가지게 된다.
- <380> 한편, 명령어타이밍 제어부(473)는 제1 리드명령어(RD0)를 셀블럭어드레스 변환시간(t_{BAT}) 및 강제프리차지시간(t_{RP})만큼 지연시킨 지연된 제1 명령어(RD0_D)를 지연부(440)를 통해 생성한다.

- <381> 이어서 첫번째 타이밍(t_0)에서 명령어타이밍 제어부(473)에서 생성된 지연된 제1 명령어(RD0_D)에 대응하는 제1 단위셀블럭(BL0)의 워드라인(WL0)이 활성화되도록 데이터 액세스 제어부(460)를 제어한다. 이어서 활성화된 워드라인(WL0)에 대응하는 4K개의 데이터가 로컬비트라인 센스앰프에 의해 감지, 증폭된다.
- <382> 이어서 감지 증폭된 4K개의 데이터는 데이터래치부(450)로 이동되어 래치된다. 이어서 래치된 4K개의 데이터중에서 제1 리드명령어(CD0)에 대응하여 입력된 컬럼어드레스에 의해 선택된 하나의 데이터(D0)가 I/O센스앰프부(도24의 1200 참조)로 출력된다. 여기서도 전술한 바와 같이 출력데이터(D0)는 두번째 타이밍(t_0)에 입출력버퍼등을 통해 메모리장치의 외부로 출력된다.
- <383> 한편, 첫번째 타이밍(t_0)중에 명령어제어부(470)는 제2 리드명령어(RD1)를 입력받아, 그에 대응하는 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환한다. 이어서 명령어제어부(470)는 제1 리드명령어(RD0)와 제2 리드명령어(RD1)에 의해 데이터가 액세스되는 단위셀블럭을 비교한다. 여기서 제1 리드명령어(RD0)와 제2 리드명령어(RD1)에 의해 데이터가 리드되는 단위셀블럭이 동일하기 때문에 제1 리드명령어(RD1)에 대응하여 감지 증폭되어 제1 단위셀블럭(BL0)의 로컬비트라인 센스앰프부에 래치된 4K개의 데이터를 강제프리차지시킨다.(도33의 'Y'구간)
- <384> 또한 'Y'구간에서 명령어타이밍 제어부는 제2 리드명령어(RD1)를 셀블럭어드레스 변환시간(t_{BAT}) 및 강제프리차지시간(t_{RP})만큼 지연시킨 지연된 제2 명령어(RD1_D)를 생성한다.
- <385> 이어서 두번째 타이밍(t_1)에서 지연된 제2 리드명령어(RD1_D)에 대응하여 제1 단위셀블럭(BL0)의 워드라인(WL1)이 활성화된다. 이어서 활성화된 워드라인(WL1)에 대응하는

4K개의 데이터가 로컬비트라인 센스앰프에 의해 감지 증폭되어, 데이터래치부(450)로 이동되어 래치된다. 이어서 데이터래치부(450)에 래치된 4K개의 데이터중에서 제2 리드 명령어(RD1)에 대응하여 입력된 컬럼어드레스에 의해 선택된 데이터(D1)가 I/O 센스앰프부로 출력된다.

<386> 한편, 이 타이밍(t1)에서 제1 리드명령어에 대응하여 활성화되는 워드라인(WL0)의 예비워드라인이 지정된 제2 단위셀블럭(BL1)의 워드라인(WL0)이 활성화되고, 데이터래치부(450)에 래치된 데이터는 제2 단위셀블럭(BL1)의 워드라인(WL0)에 대응하는 4K개의 단위셀에 이동되어 재저장된다. 이 동작에 대한 부분이 'intW0'이다.

<387> 이어서 두번째 타이밍중(t1)에 제3 리드명령어(RD2)가 명령어제어부(470)로 입력되어 그에 대응하는 논리적 셀블럭어드레스를 물리적 셀블럭어드레스로 변환된다. 이어서 명령어제어부(470)는 제2 명령어(RD1)와 제3 명령어(RD2)에 의해 데이터가 액세스되는 단위셀블럭을 비교한다. 여기서 제2 리드명령어(RD1)와 제3 리드명령어(RD2)에 의해 데이터가 액세스되는 단위셀블럭이 동일하기 때문에 제2 리드명령어(RD1)에 대응하여 감지 증폭되어 제1 단위셀블럭의 로컬비트라인 센스앰프에 래치된 4K개의 데이터를 강제프리차지시킨다.(도33의 'Z'구간)

<388> 또한 'Z'구간에서 명령어타이밍 제어부(473)는 제3 리드명령어(RD2)를 셀블럭어드레스 변환시간(tBAT) 및 강제프리차지시간(tRP)만큼 지연시킨 지연된 제3 명령어(RD2_D)를 생성한다.

<389> 세번째타이밍(t2)에서 지연부(440)를 거쳐서 셀블럭어드레스 변환시간(tBAT) 및 강제프리차지시간(tRP)만큼 지연된 제3 명령어(RD2_D)에 의해 제1 단위셀블럭(BL0)의 워드라인(WL2)이 활성화된다.

- <390> 이어서 활성화된 워드라인(WL2)에 대응하는 4K개의 데이터가 감지 증폭되고, 데이터 래치부(450)로 이동되어 래치된다. 이어서 데이터래치부(450)에 래치된 4K개의 데이터 중에서 제3 리드명령어(RD2)에 대응하여 입력된 컬럼어드레스에 의해 선택된 데이터(D2)가 I/O 센스앰프부로 출력된다.
- <391> 한편, 이 타이밍(t2)에서 제2 리드명령어(RD1)에 대응하여 활성화되는 워드라인(WL1)의 예비워드라인이 지정된 제2 단위셀블럭(BL1)의 워드라인(WL1)이 활성화된다. 이어서 데이터래치부(450)에 래치된 데이터는 제2 단위셀블럭(BL1)의 워드라인(WL1)에 대응하는 4K개의 단위셀에 이동되어 재저장된다. 이 동작에 대한 부분이 'intW1'이다.
- <392> 전술한 바와 같이 도32에 도시된 메모리 장치는 전체적인 동작은 도24에 도시된 메모리장치와 같지만, 셀블럭어드레스를 변환하고 강제프리차지동작을 수행하는데에는 직접 입력받은 리드명령어를 사용하고, 이후의 워드라인 활성화, 단위셀의 데이터를 감지 증폭, 재저장등의 데이터억세스에 관련된 동작은 셀블럭어드레스 변환시간(tBAT) 및 강제프리차지시간(tRP)만큼 지연시킨 지연된 리드명령어(RD_D)를 사용한다는 점이 다른 점이다.
- <393> 이렇게 함으로서 메모리 장치의 초기동작시, 처음 실행되는 리드명령어에 대응하는 셀블럭어드레스 변환시간(tBAT) 및 강제프리차지시간(tRP) 만큼의 레이턴시(도33의 'X')를 가지게 되지만, 두번째 실행되는 리드명령어부터는 로우사이클 타임이 셀블럭어드레스 변환시간(tBAT) 및 강제프리차지시간(tRP) 만큼이 줄어드는 효과를 가지게 된다.
- <394> 이는 이전 리드명령어에 대응하여 데이터를 감지 증폭할 때에 다음 리드명령어에 해당하는 셀블럭어드레스 변환 및 강제프리차지 동작을 하기 때문에 전체적인 로우 사이클 타임에 영향을 주지 않기 때문이다.

- <395> 이상 도32에 도시된 메모리 장치의 리드명령어에 대한 동작에 대해 살펴보았는데, 라이트동작에 관해서는 리드동작과 전체적으로 동일하기 때문에 이에 관한 설명은 생략한다. 즉, 라이트 동작시에는 데이터래치부(450)에 래치된 데이터중에서 선택된 데이터를 외부로 출력하는 동작 대신에, 외부에서 라이트 명령어에 대응하여 입력된 데이터를 데이터래치부(450)에 래치된 데이터중에서 선택된 데이터를 덮어쓰는 동작만 다르게 된다.
- <396> 도34는 본 발명의 메모리 장치의 동작을 쉽게 이해하기 위해, 도16에 도시된 메모리 장치의 동작(도22참조)과 비교하여 나타낸, 도33에 도시된 메모리 장치의 동작타이밍도이다.
- <397> 도34를 참조하여 살펴보면, 제1 명령어(CD0)를 이용하여 a_0 구간에서 셀블럭어드레스 변환 및 강제프리차지 동작을 수행하고, b_0 구간에서 제1 명령어(CD0)를 ' $t_{RP} + t_{BAT}$ ' 시간만큼 지연된 제1 명령어(CD0_D)를 이용하여 워드라인 활성화 및 활성화된 워드라인에 대응하는 데이터를 감지 증폭하게 된다. 그리고 b_0 구간중의 a_1 구간에 다음 실행될 제2 명령어(CD1)를 입력받아 셀블럭어드레스를 변환하고, 제1 명령어에 대응하는 강제프리차지 동작을 수행하게 된다. 이어서 b_1 구간에 제2 명령어(CD1)를 ' $t_{RP} + t_{BAT}$ ' 시간만큼 지연된 제2 명령어(CD1_D)를 이용하여 워드라인 활성화 및 활성화된 워드라인에 대응하는 데이터를 감지 증폭하게 된다.
- <398> 따라서 도32에 도시된 메모리 장치의 로우사이클타임은 $\text{MAX}\{0.5 \cdot t_{RC}, t_{INTW}\}$ 로 정해진다. 이 때 ' t_{RC} '는 종래의 로우사이클 타임($b_0 + b_1$)이고, ' t_{INTW} '는 실행되는 명령어에 대응하는 워드라인의 예비워드라인이 활성화되고, 데이터래치부에서 예비워드라인 지정된 단위셀블럭으로 데이터가 이동되어 재저장되는 시간을 나타낸다.

- <399> 도32에 도시된 메모리 장치의 로우사이클 타임($\text{MAX}\{0.5 \times t_{RC}, t_{INTW}\}$)은 앞에서 설명한 메모리 장치의 로우사이클 타임($\text{MAX}\{0.5 \times t_{BAT}, t_{RP} + t_{RC}, t_{INTW}\}$)에서 셀블럭어드레스 변환시간을 나타내는 'tBAT'과, 강제프리차지 타임을 나타내는 'tRP'가 제거되었다.
- <400> 통상적으로 'tINTW'보다는 'tRP'가 더 많은 시간이 소요되므로, 실질적으로 로우사이클 타임은 $0.5 \times t_{RC}$ 이 되며, 이로 인해 도32에 도시된 메모리 장치는 종래기술에 의한 메모리 장치와 대비하여 최대 1/2까지 로우사이클 타임을 줄이는 것이 가능하게 된다. 즉 도34에 도시된 바와 같이, 종래기술에 의한 메모리 장치의 로우사이클 타임은 $b_0 + b_1$ 이었어나, 본 발명에 의한 메모리 장치의 로우사이클타임은 b_0 가 되는 것이다.
- <401> 본 발명에 의해 캐패시터를 단위메모리 소자로 사용하는 메모리 장치에서는 필연적으로 데이터 액세스시간에 많은 부분을 차지하던 데이터 재저장시간에 구애받지 않고, 고속으로 데이터를 액세스할 수 있는 메모리 장치를 구현할 수 있게 되었다.
- <402> 본 발명의 기술 사상은 상기 바람직한 실시예에 따라 구체적으로 기술되었으나, 상기한 실시예는 그 설명을 위한 것이며 그 제한을 위한 것이 아님을 주의하여야 한다. 또한, 본 발명의 기술 분야의 통상의 전문가라면 본 발명의 기술 사상의 범위내에서 다양한 실시예가 가능함을 이해할 수 있을 것이다.

【발명의 효과】

- <403> 본 발명에 의해서 데이터의 재저장시간만큼 데이터 액세스시간이 감소되어 고속으로 데이터 액세스가 가능한 메모리 장치를 구현 할 수 있다.

<404> 본 발명의 의한 메모리 장치는 동일 셀블럭에서 연속적인 데이터를 액세스하든지, 또는 서로 다른 셀블럭 간에 교대로 데이터를 액세스하든지, 데이터 액세스 패턴에 상관없이 항상 고속으로 데이터를 연속해서 액세스할 수 있다. 따라서 본 발명의 메모리 장치를 메인메모리로 구비한 시스템은 메모리 장치의 데이터 액세스 패턴에 상관없이 전체적인 시스템 속도가 크게 향상되는 효과를 기대할 수 있다.

<405> 또한 본 발명에 의한 메모리 장치는 종래의 메모리 장치 구조를 최대한으로 유지시키면서 태그관련 블럭 몇개만을 추가함으로써 고속 데이터 액세스를 구현했기 때문에, 제조개발 비용을 최대한 줄이면서도 고속으로 동작하는 메모리 장치를 구현할 수 있다.

【특허청구범위】**【청구항 1】**

다수개의 단위셀을 구비하는 제1 셀블럭;

다수개의 단위셀을 구비하는 제2 셀블럭; 및

상기 제1 셀블럭에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 상기 제1 셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 제2 셀블럭에서는 상기 제1 데이터의 재저장 동작이 수행되도록 제어하기 위한 제어수단

을 구비하는 메모리 장치.

【청구항 2】

제 1 항에 있어서,

상기 제1 데이터를 상기 제1 셀블럭으로부터 전달받아 래치하기 위한 래치 수단;

및

상기 래치수단에 래치된 상기 제1 데이터를 상기 제2 셀블럭으로 전달하기 위한 신호라인을 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 3】

제 2 항에 있어서

상기 제1 및 제2 셀블럭은,

상기 제1 데이터 또는 상기 제2 데이터를 감지 증폭하기 위한 비트라인 센스앰프를 각각 구비하고,

상기 래치수단은 상기 비트라인 센스앰프에서 감지 증폭되는 데이터를 래치하는 것을 특징으로 하는 메모리 장치.

【청구항 4】

제 3 항에 있어서,

상기 신호라인을 상기 제1 및 제2 셀블럭에 구비된 비트라인 센스앰프와 선택적으로 연결하기 위한 연결부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 5】

제 3 항에 있어서,

상기 제어수단은

상기 제1 셀블럭과 상기 제2 셀블럭에서 상기 제1 데이터와 상기 제2 데이터가 교대로 액세스될 때, 상기 래치수단에 래치된 제1 데이터를 상기 제1 셀블럭에 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 상기 제2 셀블럭에서 상기 제2 데이터가 액세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 6】

제 5 항에 있어서,

상기 제어수단은

상기 래치수단에 래치된 데이터가 실행중인 리드명령어에 대응하는 출력데이터가 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 7】

다수의 단위셀을 구비하는 단위셀블럭과, 상기 단위셀블럭의 일측에 구비되어 상기 단위셀블럭의 일부 데이터를 감지하여 증폭하기 위한 제1 로컬비트라인 센스앰프부와, 상기 단위셀블럭의 타측에 구비되어 상기 단위셀블럭의 나머지 데이터를 감지하여 증폭하기 위한 제2 로컬비트라인 센스앰프를 구비하는 제1 기본셀블럭;

상기 제1 기본셀블럭과 같은 구성을 가지는 제2 내지 제4 기본셀블럭;

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제1 로컬비트라인 센스앰프부에서 감지 증폭된 데이터를 래치하기 위한 제1 글로벌비트라인 센스앰프부;

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제2 로컬비트라인 센스앰프부에서 감지 증폭된 데이터를 래치하기 위한 제2 글로벌비트라인 센스앰프부;

상기 제1 기본셀블럭의 제2 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부로 연결하거나, 상기 제2 기본셀블럭의 제1 로컬비트라인 센스앰프부를 상기 제2 글로벌 비트라인 센스앰프부와 선택적으로 연결하기 위한 제1 글로벌비트라인 연결부;

상기 제3 기본셀블럭의 제2 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부로 연결하거나, 상기 제4 기본셀블럭의 제1 로컬비트라인 센스앰프부를 상기

제1 글로벌비트라인 센스앰프부와 선택적으로 연결하기 위한 제2 글로벌비트라인
연결부; 및

제1 글로벌비트라인 센스앰프부와 상기 제2 글로벌비트라인 센스앰프부에 래치된
데이터를 재저장 동작에 사용되도록 제어하기 위한 제어수단
을 구비하는 메모리 장치.

【청구항 8】

제 7 항에 있어서,

상기 제어수단은

상기 제1 기본셀블럭에 연속적으로 제1 및 제2 데이터가 액세스될 때, 상기 제1 기
본셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세
스되도록 제어하고, 상기 제1 기본셀블럭 이외의 다른 기본셀블럭에서 상기 제1 데이터
의 재저장 동작이 수행되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 9】

제 8 항에 있어서,

상기 제어수단은

상기 제1 기본셀블럭과 상기 제2 기본셀블럭에서 제1 데이터 및 제2 데이터가 교대
로 액세스 될 때, 상기 제1 데이터를 제1 기본셀블럭에 재저장시키고, 상기 재저장 타이

밍과 실질적으로 동일한 타이밍에 상기 제2 기본셀블럭에서 상기 제2 데이터가 액세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 10】

제 9 항에 있어서,

상기 제어수단은

상기 제1 및 제2 글로벌비트라인 센스앰프부에 래치된 데이터중에서 선택된 데이터를 실행중인 리드명령어에 대응하는 출력데이터로 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 11】

제 8 항에 있어서,

상기 제1 기본셀블럭과 같은 구성을 가지는 상기 제5 기본셀블럭; 및

상기 제5 기본셀블럭의 제2 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부와 연결하기 위한 제3 글로벌비트라인 연결부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 12】

제 11 항에 있어서,

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제1 로컬비트라인 센스앰프부에 의해 감지 증폭된 데이터를 상기 제1 글로벌비트라인 센스앰프부와 연결 -상기 제1 내지 제3 글로벌 비트라인 연결부중 하나를 통해서 연결되거나 또는 직접연결- 하거나,

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제2 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부와 연결 -상기 제1 내지 제3 글로벌 비트라인 연결부중 하나를 통해서 연결되거나 또는 직접연결- 하기 위한 글로벌비트라인을 더 구비한 것을 특징으로 하는 메모리 장치.

【청구항 13】

제 12 항에 있어서,

상기 제1 내지 제5 기본셀블럭에 각각 구비된 제1 및 제2 로컬비트라인 센스앰프부는,

상기 제1 내지 제3 글로벌비트라인 연결부중 하나와 연결되거나 또는 상기 글로벌 비트라인과 연결되기 위한 스위칭수단을 각각 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 14】

다수의 단위셀을 구비하는 단위셀블럭과, 상기 단위셀블럭의 일측에 구비되어 상기 단위셀블럭의 일부 데이터를 감지하여 증폭하기 위한 제1 로컬비트라인 센스앰프부와,

상기 단위셀블럭의 타측에 구비되어 상기 단위셀블럭의 나머지 데이터를 감지하여 증폭하기 위한 제2 로컬비트라인 센스앰프를 구비하는 제1 기본셀블럭;

상기 제1 기본셀블럭과 같은 구성을 가지는 제2 내지 제4 기본셀블럭;

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제1 로컬비트라인 센스앰프부에서 감지 증폭된 데이터를 래치하기 위한 제1 글로벌비트라인 센스앰프부;

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제2 로컬비트라인 센스앰프부에서 감지 증폭된 데이터를 래치하기 위한 제2 글로벌비트라인 센스앰프부; 및

제1 글로벌비트라인 센스앰프부와 상기 제2 글로벌비트라인 센스앰프부에 래치된 데이터를 재저장 동작에 사용되도록 제어하기 위한 제어수단

을 구비하는 메모리 장치.

【청구항 15】

제 14 항에 있어서,

상기 제어수단은

상기 제1 기본셀블럭에 연속적으로 제1 및 제2 데이터가 액세스될 때, 상기 제1 기본셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 제1 기본셀블럭 이외의 다른 기본셀블럭에서 상기 제1 데이터의 재저장 동작이 수행되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 16】

제 15 항에 있어서,

상기 제어수단은

상기 제1 기본셀블럭과 상기 제2 기본셀블럭에서 제1 데이터 및 제2 데이터가 교대로 액세스 될 때, 상기 제1 데이터를 제1 기본셀블럭에 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 상기 제2 기본셀블럭에서 상기 제2 데이터가 액세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 17】

제 16 항에 있어서,

상기 제어수단은

상기 제1 및 제2 글로벌비트라인 센스앰프부에 래치된 데이터중에서 선택된 데이터를 실행중인 리드명령어에 대응하는 출력데이터로 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 18】

제 17 항에 있어서,

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제1 로컬비트라인 센스앰프부에 의해 감지 증폭된 데이터를 상기 제1 글로벌비트라인 센스앰프부와 연결하기 위한 제1 글로벌비트라인; 및

상기 제1 내지 제4 기본셀블럭에 각각 구비된 제2 로컬비트라인 센스앰프부를 상기 제2 글로벌비트라인 센스앰프부와 연결하기 위한 제2 글로벌비트라인을 더 구비한 것을 특징으로 하는 메모리 장치.

【청구항 19】

제 18 항에 있어서,

상기 제1 내지 제5 기본셀블럭에 각각 구비된 제1 로컬비트라인 센스앰프부는 상기 제1 글로벌비트라인과 연결되기 위한 제1 스위칭수단을 구비하고,

상기 제1 내지 제5 기본셀블럭에 각각 구비된 제2 로컬비트라인 센스앰프부는 상기 제2 글로벌비트라인과 연결되기 위한 제2 스위칭수단을 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 20】

다수의 단위셀을 각각 구비하고, 상기 다수의 단위셀에 저장된 데이터를 감지증폭하기 위한 로컬비트라인 센스앰프부를 각각 구비하는 다수의 기본셀블럭;

상기 기본셀블럭에 각각 구비된 다수의 로컬비트라인 센스앰프부에 의해 감지 증폭된 데이터를 래치하기 위한 글로벌비트라인 센스앰프부; 및

상기 래치된 데이터를 재저장 동작에 사용되도록 제어하기 위한 제어수단을 구비하는 메모리 장치.

【청구항 21】

제 20 항에 있어서,

상기 다수의 로컬비트라인 센스앰프부는 상기 글로벌비트라인 센스앰프부와 선택적으로 연결되기 위해 스위칭 수단을 각각 구비하는 특징으로 하는 메모리 장치.

【청구항 22】

제 21 항에 있어서,

상기 제어수단은

상기 다수의 기본셀블럭중에서 선택된 제1 기본셀블럭에 연속적으로 제1 및 제2 데이터가 액세스될 때 상기 제1 기본셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 제1 기본셀블럭 이외의 다른 기본셀블럭에서 상기 제1 데이터의 재저장 동작이 수행되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 23】

제 22 항에 있어서,

상기 제어수단은

상기 제1 기본셀블럭과 상기 제2 기본셀블럭에서 제1 데이터 및 제2 데이터가 교대로 액세스 될 때, 상기 제1 데이터를 상기 제1 기본셀블럭에 재저장시키고, 상기 재저장

타이밍과 실질적으로 동일한 타이밍에 상기 제2 기본셀블럭에서 상기 제2 데이터가 역 세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 24】

제 23 항에 있어서,

상기 제어수단은

상기 글로벌비트라인 센스앰프부에 래치된 데이터가 현재 실행중인 리드명령어에 대응하는 출력데이터로 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 25】

제 24 항에 있어서,

상기 다수의 로컬비트라인 센스앰프부와 상기 글로벌비트라인 센스앰프부를 연결하기 위한 글로벌 비트라인을 더 구비하는 특징으로 하는 메모리 장치.

【청구항 26】

제 21 항에 있어서,

상기 글로벌비트라인 센스앰프부는

상기 로컬비트라인 센스앰프부에 의해 감지 증폭된 일부 데이터를 래치하기 위한 제1 글로벌비트라인 센스앰프부; 및

로컬비트라인 센스앰프부에 의해 감지 증폭된 나머지 데이터를 래치하기 위한 제2 글로벌비트라인 센스앰프부를 구비하는 메모리 장치.

【청구항 27】

제 26 항에 있어서,

상기 제어수단은

상기 다수의 기본셀블럭중에서 선택된 제1 기본셀블럭에 연속적으로 제1 및 제2 데이터가 액세스될 때 상기 제1 기본셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 제1 기본셀블럭 이외의 다른 기본셀블럭에서 상기 제1 데이터의 재저장 동작이 수행되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 28】

제 27 항에 있어서,

상기 제어수단은

상기 제1 기본셀블럭과 상기 제2 기본셀블럭에서 제1 데이터 및 제2 데이터가 교대로 액세스 될 때, 상기 제1 데이터를 상기 제1 기본셀블럭에 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 상기 제2 기본셀블럭에서 상기 제2 데이터가 액세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 29】

제 28 항에 있어서,

상기 제어수단은

상기 제1 및 제2 글로벌비트라인 센스앰프부에 래치된 데이터가 현재 실행중인 리드명령어에 대응하는 출력데이터로 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 30】

제 28 항에 있어서,

상기 다수의 로컬비트라인 센스앰프부와 상기 제1 및 제2 글로벌비트라인 센스앰프부를 연결하기 위한 글로벌 비트라인을 더 구비하는 특징으로 하는 메모리 장치.

【청구항 31】

각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 N개로 구성된 단위셀블럭에, 추가적으로 M개의 워드라인을 가지는 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭; 및

상기 N+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭으로부터 액세스되는 데이터를 상기 제1 단위셀블럭 또는 상기 제2 단위셀블럭으로 재저장시키도록 제어하는 제어수단

을 구비하는 메모리 장치.

【청구항 32】

제 31 항에 있어서,

상기 제어수단은

상기 입력되는 로우어드레스에 대응하여 2개의 단위셀블럭에서 각각 워드라인이 활성화되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 33】

제 32 항에 있어서,

상기 2개의 단위셀블럭에 각각 활성화되는 워드라인을 선택하기 위한 각각의 어드레스는 동일 어드레스인 것을 특징으로 하는 메모리 장치.

【청구항 34】

제 32 항에 있어서,

상기 제어수단은

상기 제1 단위셀블럭에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 상기 제1 단위셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고, 상기 제2 단위셀블럭에서는 상기 제1 데이터의 재저장동작이 수행하도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 35】

제 34 항에 있어서,

상기 제어수단은

상기 제1 데이터 및 상기 제2 데이터가 상기 제1 및 제2 단위셀블럭에서 교대로 액세스될 때, 상기 제1 데이터를 상기 제1 단위셀블럭에 재저장시키고, 상기 재저장 타이밍과 실질적으로 동일한 타이밍에 상기 제2 단위셀블럭에서 상기 제2 데이터가 액세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 36】

제 35 항에 있어서,

상기 셀블럭은

상기 N+1개의 단위셀블럭에 각각 구비된 로컬비트라인 센스앰프부에서 감지 증폭되는 데이터를 래치하기 위한 글로벌비트라인 센스앰프부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 37】

제 36 항에 있어서,

상기 N+1개의 단위셀블럭에 각각 구비된 로컬비트라인 센스앰프부와 상기 글로벌비트라인 센스앰프부를 선택적으로 연결하기 위한 글로벌비트라인 연결부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 38】

제 36 항에 있어서,

상기 N+1개의 단위셀블럭에 각각 구비된 상기 로컬비트라인 센스앰프부와 상기 글로벌비트라인 센스앰프부를 연결하기 위한 글로벌비트라인을 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 39】

제 36 항에 있어서,

상기 제어수단은

상기 글로벌비트라인 센스앰프부에 래치된 데이터가 현재 실행중인 리드명령어에 대응하는 출력데이터로 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 40】

각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 M개의 워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성되는 셀블럭;

상기 (N+1) 개의 워드라인 중에서, 적어도 어느 한 워드라인을 예비워드라인으로 할지의 여부에 대한 정보를 저장하기 위한 예비셀블럭 테이블;

상기 로우어드레스를 입력받아 N개의 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스를 감지하고, 이를 상기 N+1개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하여 출력하기 위한 태그블럭; 및

상기 물리적 셀블럭어드레스에 의해 선택된 단위셀블럭에서의 한 워드라인과, 상기 워드라인에 대한 -상기 예비셀블럭 테이블로부터 제공되는 정보에 의해 정해지는- 예비워드라인을 활성화시키기 위해 상기 태그블럭과 상기 예비셀블럭 테이블을 제어하는 제어수단

을 구비하는 메모리 장치.

【청구항 41】

제 40 항에 있어서,

상기 제어수단은

상기 N+1개의 단위셀블럭중에서 선택된 하나의 단위셀블럭에 연속적으로 제1 및 제2 데이터가 액세스될 때, 상기 제1 데이터에 대한 재저장동작은 선택된 예비워드라인을 통해 이루어지도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 42】

제 41 항에 있어서

상기 예비워드라인은 M개인 것을 특징으로 하는 메모리 장치.

【청구항 43】

제 42 항에 있어서,

상기 태그블럭은

N+1 개의 단위셀블럭에 각각 구비된 M개의 워드라인이 어떤 논리적셀블럭에 대응되는 지를 각각 저장하고 있는 N+1개의 단위태그테이블;

상기 로우어드레스에서 감지된 논리적 셀블럭어드레스와, 로컬어드레스(단위셀블럭의 한 워드라인을 선택하기 위한 어드레스)에 대응되어 상기 N+1개의 단위태그테이블에서 출력되는 정보 -상기 로컬어드레스에 의해 선택된 워드라인이 어떤 논리적 단위셀블럭에 대응되는 것인지에 관한 N+1개의 데이터 정보- 를 각각 비교하기 위한 N+1개의 비교수단;

상기 N+1개의 비교수단에서 비교한 정보를 인코딩하여 상기 물리적 셀블럭어드레스를 출력하기 위한 인코딩수단; 및

상기 N+1개의 단위태그테이블과 상기 N+1개의 비교수단과 상기 인코딩수단을 제어하기 위한 태그제어부를 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 44】

제 43 항에 있어서,

상기 태그블럭은,

상기 예비셀블럭 테이블에서 출력되는 정보 -상기 로컬어드레스에 의해 선택된 워드라인에 대응하는 예비워드라인이 상기 N+1개의 단위셀블럭중에서 어떤 단위셀블럭에 있는지에 대한 정보- 를 디코딩하여, 상기 N+1개의 단위태그테이블중에서 선택된 하나의 단위태그테이블로 출력하기 위한 셀블럭어드레스 디코딩부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 45】

제 44 항에 있어서,

상기 예비셀블럭 테이블은

상기 M개의 예비워드라인이 상기 N+1개의 물리적 단위셀블럭중에서 어떤 단위셀블럭에 있는지에 대한 정보를 저장하기 위한 M개의 레지스터를 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 46】

제 45 항에 있어서,

상기 제어수단은

상기 N+1단위셀블럭중에서 선택된 제1 단위셀블럭에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 상기 제1 단위셀블럭에서는 상기 제1 데이터의 재저장 동작을 수행하지 않고 상기 제2 데이터가 액세스되도록 제어하고,

상기 제1 단위셀블럭이외의 다른 제2 단위셀블럭에서는 상기 제1 데이터의 재저장 동작이 수행하도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 47】

제 46 항에 있어서,

상기 제어수단은

상기 제1 및 제2 단위셀블럭에 제1 데이터 및 제2 데이터가 교대로 액세스 될 때, 상기 제1 단위셀블럭에 상기 제1 데이터를 재저장시키는 타이밍과 실질적으로 동일한 타이밍에 상기 제2 단위셀블럭에서 상기 제2 데이터가 액세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 48】

제 47 항에 있어서,

상기 셀블럭은

상기 N+1개의 단위셀블럭에 각각 구비된 로컬비트라인 센스앰프부에서 감지 증폭되는 데이터를 래치하기 위한 데이터래치부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 49】

제 48 항에 있어서,

상기 제어수단은

상기 데이터래치부에 래치된 데이터가 실행중인 리드명령어에 대응하는 출력데이터로 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 50】

제 48 항에 있어서,

상기 셀블럭은

상기 N+1개의 단위셀블럭에 각각 구비된 로컬비트라인 센스앰프부와 상기 데이터래치부를 선택적으로 연결하기 위한 글로벌비트라인 연결부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 51】

제 48 항에 있어서,

상기 셀블럭은

상기 N+1개의 단위셀블럭에 각각 구비된 상기 로컬비트라인 센스앰프부와 상기 데이터래치부를 연결하기 위한 글로벌비트라인을 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 52】

각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 M개의 워드라인을 더 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭;

상기 N+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭으로부터 액세스되는 데이터가 상기 제1 단위셀블럭 또는 제2 단위셀블럭으로 재저장되도록 제어하는 데이터엑세스 제어수단; 및

현재 실행되는 제1 명령어에 대응하는 데이터 액세스 중에, 다음 실행될 제2 명령어에 대응하는 셀블럭어드레스 변환동작과 상기 제1 명령어에 대응하는 비트라인에 대한 강제프리차지동작이 이루어지도록 상기 데이터엑세스 제어수단을 제어하는 명령어 제어수단

을 구비하는 메모리 장치.

【청구항 53】

제 52 항에 있어서,

상기 데이터엑세스 제어수단은

상기 제1 단위셀블럭에 대해 연속적으로 제1 데이터 및 제2 데이터가 액세스될 때, 상기 제1 단위셀블럭에서는 상기 제1 데이터의 재저장동작을 수행하지 않고 상기 제2 데

이터가 액세스되도록 제어하고, 상기 제2 단위셀블럭에서는 상기 제1 데이터의 재저장 동작이 수행하도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 54】

제 53 항에 있어서,

상기 데이터엑세스 제어수단은

상기 제1 및 제2 단위셀블럭에 제1 데이터 및 제2 데이터가 교대로 액세스 될 때, 상기 제1 단위셀블럭에 상기 제1 데이터를 재저장시키는 타이밍과 실질적으로 동일한 타이밍에 상기 제2 단위셀블럭에서 상기 제2 데이터가 액세스되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 55】

제 52 항에 있어서,

상기 명령어 제어수단은,

상기 제1 명령어에 대응하여 입력된 로우어드레스에서 논리적 셀블럭어드레스를 감지하여, N+1개의 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하는 셀블럭어드레스 변환수단;

상기 제1 명령어에 의해 감지, 증폭된 데이터를 강제프리차지시키기 위한 강제프리차지 제어부; 및

상기 제1 명령어에 대응하여 감지증폭된 데이터가 래치된 이후에 상기 제1 명령어에 대응하는 상기 강제프리차지 동작이 수행되도록 상기 강제프리차지 제어부를 제어하는 명령어 타이밍제어부를 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 56】

제 55 항에 있어서,

상기 명령어 타이밍제어부는

상기 제1 명령어에 대응하는 상기 강제프리차지 동작 및 상기 제2 명령어에 대응하는 상기 셀블럭어드레스 변환동작을 수행하는 시간만큼 상기 제2 명령어를 지연시켜 출력하기 위한 지연수단을 구비하고,

상기 제2 명령어에 대응하는 데이터의 감지, 증폭동작은 상기 지연수단에 의해 지연된 제2 명령어에 의해 수행되도록 상기 데이터억세스 제어수단을 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 57】

제 56 항에 있어서,

상기 셀블럭은

상기 N+1개의 단위셀블럭에 각각 구비된 로컬비트라인 센스앰프부에서 감지 증폭되는 데이터를 래치하기 위한 데이터래치부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 58】

제 57 항에 있어서,

상기 제어수단은

상기 데이터래치부에 래치된 데이터가 실행중인 리드명령어에 대응하는 출력데이터로 되도록 제어하는 것을 특징으로 하는 메모리 장치.

【청구항 59】

제 57 항에 있어서,

상기 셀블럭은

상기 N+1개의 단위셀블럭에 각각 구비된 로컬비트라인 센스앰프부와 상기 데이터래치부를 선택적으로 연결하기 위한 글로벌비트라인 연결부를 더 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 60】

제 57 항에 있어서,

상기 셀블럭은

상기 N+1개의 단위셀블럭에 각각 구비된 상기 로컬비트라인 센스앰프부와 상기 데이터래치부를 연결하기 위한 글로벌비트라인을 구비하는 것을 특징으로 하는 메모리 장치.

【청구항 61】

다수개의 단위셀을 각각 구비하는 제1 및 제2 셀블럭을 구비하는 메모리 장치의 구동방법에 있어서,

상기 제1 셀블럭에서 제1 데이터를 감지, 증폭하는 제1 단계;

상기 제2 셀블럭에서 상기 제1 데이터를 재저장하는 제2 단계; 및

상기 제1 셀블럭에서 제2 데이터를 감지, 증폭하는 제3 단계를 포함하며, 상기 제2 단계 및 상기 제3 단계는 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법.

【청구항 62】

제 61 항에 있어서,

상기 제2 단계는

상기 제1 데이터를 이동시켜 래치하는 제4 단계; 및

래치된 상기 제1 데이터를 상기 제2 셀블럭으로 이동시켜 재저장하는 제5 단계를 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 63】

제 62 항에 있어서,

상기 제4 단계에서 래치된 제1 데이터를 입력된 리드명령어에 대응하는 출력데이터로 출력시키는 제6단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 64】

각각 M개의 워드라인을 가지며, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭을 구비하는 메모리 장치의 구동방법에 있어서,

상기 N+1개의 단위셀블럭중에서 선택된 제1 단위셀블럭에서 제1 데이터를 감지, 증폭하는 제1 단계;

상기 N+1개의 단위셀블럭중에서 선택된 제2 단위셀블럭으로 상기 제1 데이터를 재저장하는 제2 단계; 및

상기 제1 셀블럭에서 제2 데이터를 감지, 증폭하는 제3 단계를 포함하며, 상기 제2 단계 및 상기 제3 단계는 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법.

【청구항 65】

제 64 항에 있어서,

상기 제2 단계는

상기 제1 데이터를 이동시켜 래치하는 제4 단계; 및

래치된 상기 제1 데이터를 상기 제2 셀블럭으로 이동시켜 재저장하는 제5 단계를 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 66】

제 65 항에 있어서,

상기 제4 단계에서 래치된 제1 데이터를 입력된 리드명령어에 대응하는 출력데이터로 출력시키는 제6단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 67】

제 66 항에 있어서,

상기 제1 데이터가 래치된 후에 상기 감지, 증폭된 제1 데이터를 강제프리차지시켜 제거하는 제7 단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 68】

각각 M개의 워드라인을 가지고, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭을 구비하는 메모리 장치의 구동방법에 있어서,

상기 N+1개의 단위셀블럭 중에서 선택된 제1 단위셀블럭의 제1 워드라인을 활성화시키는 제1 단계;

상기 제1 워드라인에 대응되는 K개의 데이터를 감지 증폭하는 제2 단계;

감지증폭된 상기 제1 워드라인에 대응되는 K개의 데이터를 상기 제1 워드라인에 대응하는 예비워드라인이 구비된 셀블럭으로 이동시켜 재저장하는 제3 단계;

상기 제1 단위셀블럭에 제2 워드라인을 활성화시키는 제4 단계; 및

상기 제2 워드라인에 대응하는 K개의 데이터를 감지증폭하는 제5 단계를 포함하며, 제3 단계 내지 제5 단계의 수행은 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법.

【청구항 69】

제 68 항에 있어서,

상기 제 3 단계는

상기 제1 워드라인에 대응하는 K개의 데이터를 이동시켜 래치시키는 제6 단계;

상기 제1 워드라인에 대응하는 예비워드라인을 활성화시키는 제7 단계; 및

상기 래치된 K개의 데이터를 상기 예비워드라인에 대응되는 K개의 단위셀에 재저장하는 제8 단계를 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 70】

제 69 항에 있어서,

상기 제6 단계에서 래치된 K개의 데이터중 선택된 하나를 입력된 리드명령어에 대응하는 출력데이터로 출력시키는 제8단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 71】

제 70 항에 있어서,

상기 제1 워드라인에 대응하여 감지, 증폭되어 비트라인에 인가된 K개의 데이터를 강제프리차지시켜 제거하는 제9 단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 72】

각각 M개의 워드라인을 가지고, 입력되는 로우어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 상기 M개의 예비워드라인을 가지기 위해 추가의 단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭을 구비하는 메모리 장치의 구동방법에 있어서,

상기 로우어드레스를 입력받아 N개의 논리적 단위셀블럭을 선택하기 위한 논리적 셀블럭어드레스와, 선택된 단위셀블럭에 구비된 M개의 워드라인중 하나를 선택하기 위한 로컬어드레스로 감지하는 제1 단계;

상기 논리적 셀블럭어드레스를 N+1개의 물리적 단위셀블럭중 하나를 선택하기 위한 물리적 셀블럭어드레스로 변환하는 제2 단계;

상기 변환된 물리적 셀블럭어드레스에 대응하여 선택된 제1 단위셀블럭에서 상기 로컬어드레스에 대응하는 제1 워드라인을 활성화시키는 제3 단계;

상기 제1 워드라인에 대응되는 제1 데이터를 감지증폭하는 제4 단계;

상기 제1 데이터를 상기 제1 워드라인에 대응되는 예비워드라인이 지정된 제2 단위셀블럭으로 이동시켜 재저장하는 제5 단계;

다음명령을 위해 입력된 로컬어드레스에 대응하는 상기 제1 단위셀블럭의 제2 워드라인을 활성화시키는 제6 단계; 및

상기 제2 워드라인에 대응하는 제2 데이터를 감지증폭하는 제7 단계를 포함하며, 상기 제5 단계 내지 상기 제7 단계는 실질적으로 동일한 타이밍에 이루어지는 메모리 장치의 구동방법.

【청구항 73】

제 72 항에 있어서,

상기 제5 단계는

상기 제1 워드라인에 대응하는 제1 데이터를 이동시켜 래치시키는 제8 단계;

상기 제1 워드라인에 대응하는 예비워드라인을 활성화시키는 제9 단계; 및

상기 래치된 제1 데이터를 상기 예비워드라인에 대응되는 단위셀에 재저장하는 제 10 단계를 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 74】

제 73 항에 있어서,

상기 제8 단계에서 래치된 제1 데이터를 입력된 리드명령어에 대응하는 출력데이터로 출력시키는 제11단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 75】

제 74 항에 있어서,

상기 제1 워드라인에 대응하여 감지, 증폭되어 비트라인에 인가된 제1 데이터를 강제프리차지시켜 제거하는 제11 단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 76】

제 75 항에 있어서,

상기 제1 워드라인과 상기 예비워드라인은 동일한 로우어드레스에 의해 선택되는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 77】

각각 M개의 워드라인을 가지고, 입력되는 논리적 셀블럭어드레스에 대응하도록 구비된 N개의 단위셀블럭에, 추가적으로 상기 M개의 예비워드라인을 가지기 위해 추가의

단위셀블럭을 더 포함하여 N+1개의 단위셀블럭으로 구성된 셀블럭을 구비하는 메모리 장치의 구동방법에 있어서,

제 1 명령어에 대응하는 제1 논리적 셀블럭어드레스를 입력받아 상기 N+1개의 단위 셀블럭중 하나를 선택하기 위한 제1 물리적 셀블럭어드레스로 변환하는 제1 단계;

상기 제1 물리적 셀블럭어드레스에 대응하는 제1 단위셀블럭에서 제1 데이터를 감지증폭하는 제2 단계;

상기 제1 데이터를 이동시켜 래치하는 제3 단계;

상기 제1 단위셀블럭에서 감지증폭된 상기 제1 데이터를 강제프리차지시키는 제4 단계;

제 2 명령어에 대응하는 제2 논리적 셀블럭어드레스를 입력받아 제2 물리적 셀블럭어드레스로 변환하는 제5 단계;

상기 래치된 제1 데이터를 상기 N+1개의 단위셀블럭중에서 선택된 제2 단위셀블럭으로 이동시켜 재저장하는 제6 단계; 및

상기 제2 물리적 셀블럭어드레스에 대응하는 제2 단위셀블럭에서 제2 데이터를 감지증폭하는 제7 단계를 포함하며, 상기 제6 단계 및 상기 제7 단계는 실질적으로 같은 타이밍에 이루어지는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 78】

제 77 항에 있어서,

상기 제4 단계 및 상기 제5 단계를 수행하는 시간만큼 제2 명령어를 지연시켜 입력받는 제8 단계를 더 포함하고,

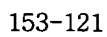
상기 제7 단계는 상기 제8 단계에 의해 지연된 제2 명령어를 이용하여 수행되는 것을 특징으로 하는 메모리 장치의 구동방법.

【청구항 79】

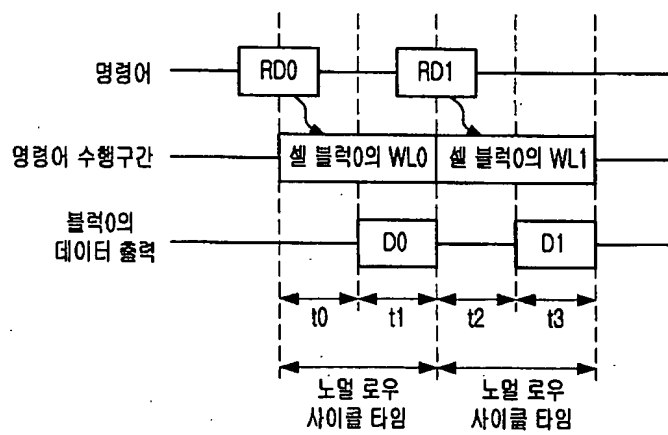
제 78 항에 있어서,

상기 제3 단계에서 래치된 제1 데이터를 입력된 리드명령어에 대응하는 출력데이터로 출력시키는 제9 단계를 더 포함하는 것을 특징으로 하는 메모리 장치의 구동방법.

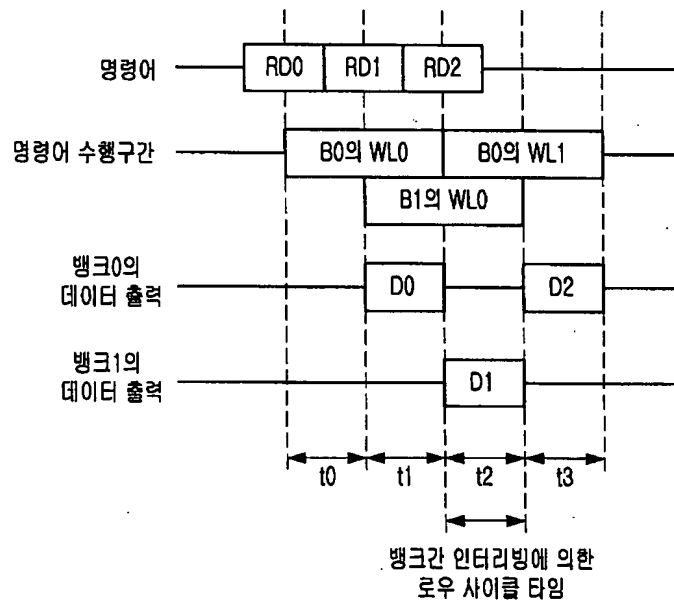
【도 1】



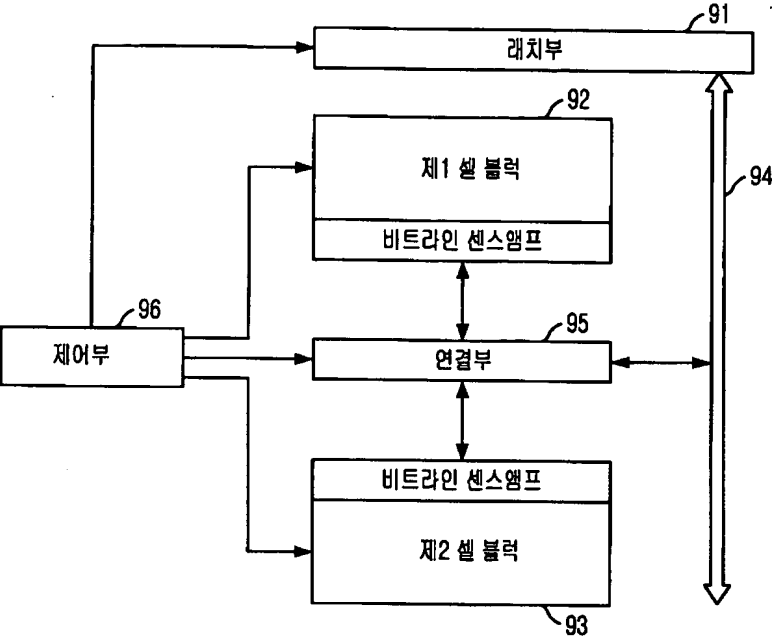
【도 3】



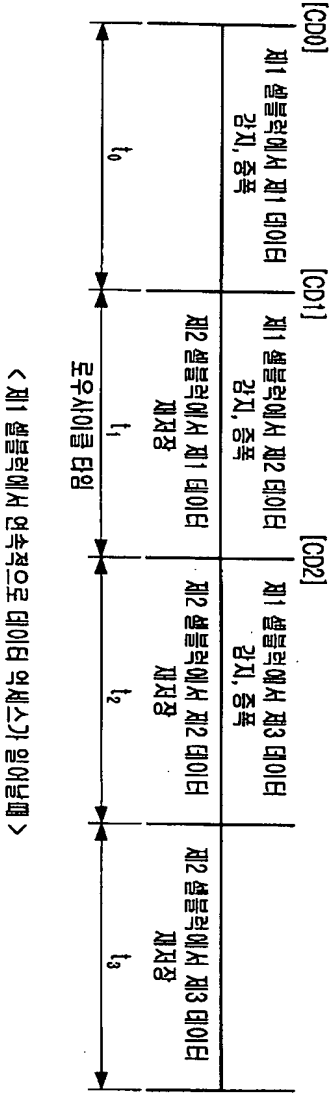
【도 4】



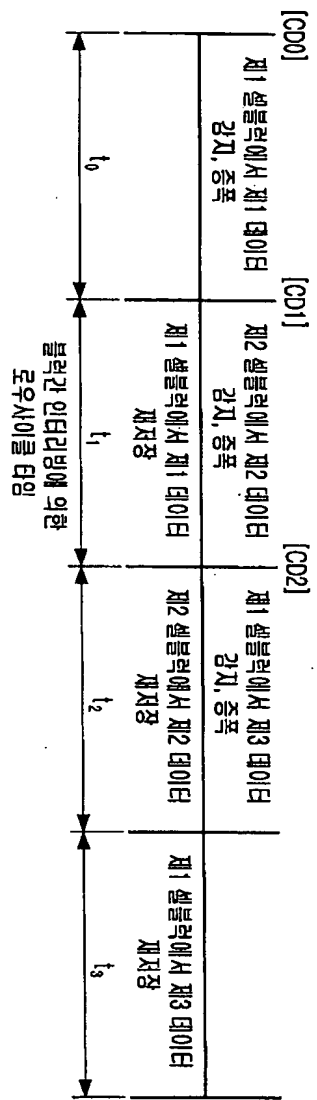
【도 5】



【도 6】

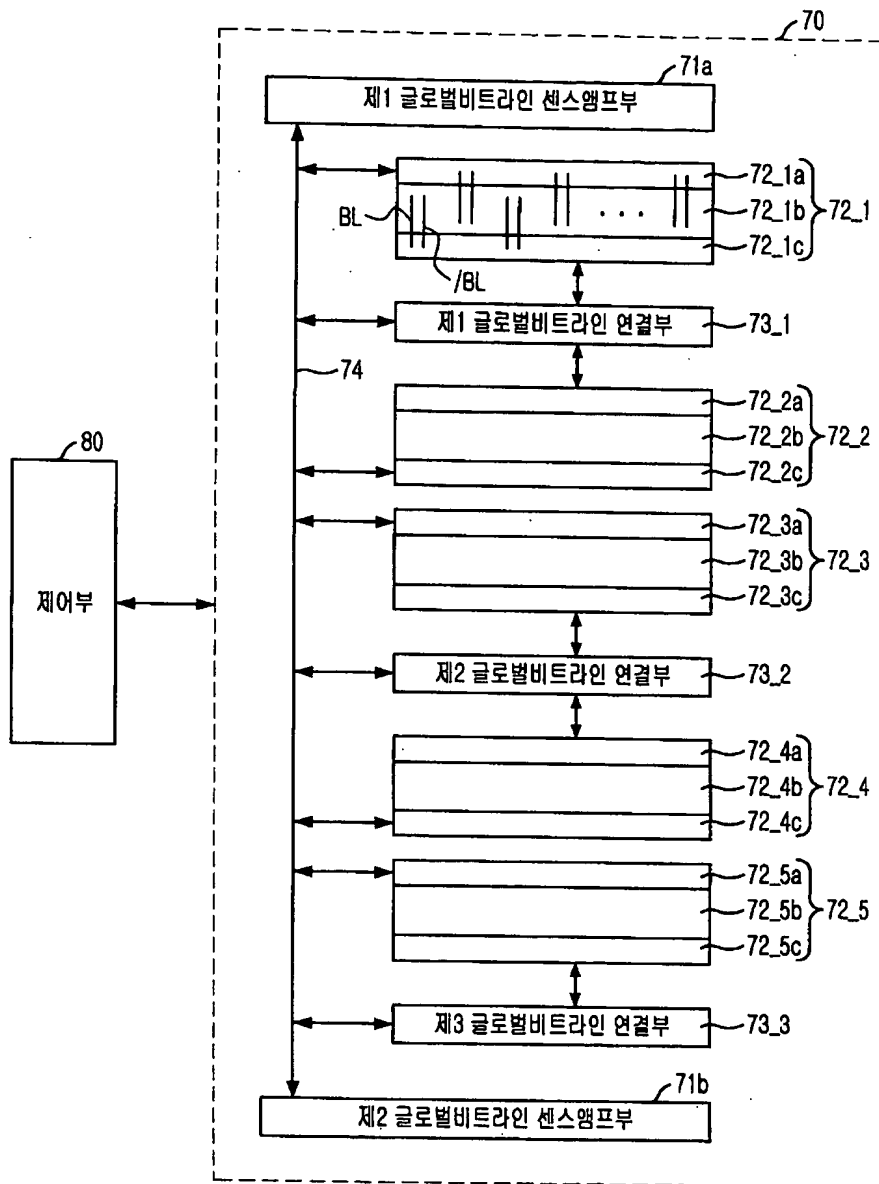


【도 7】

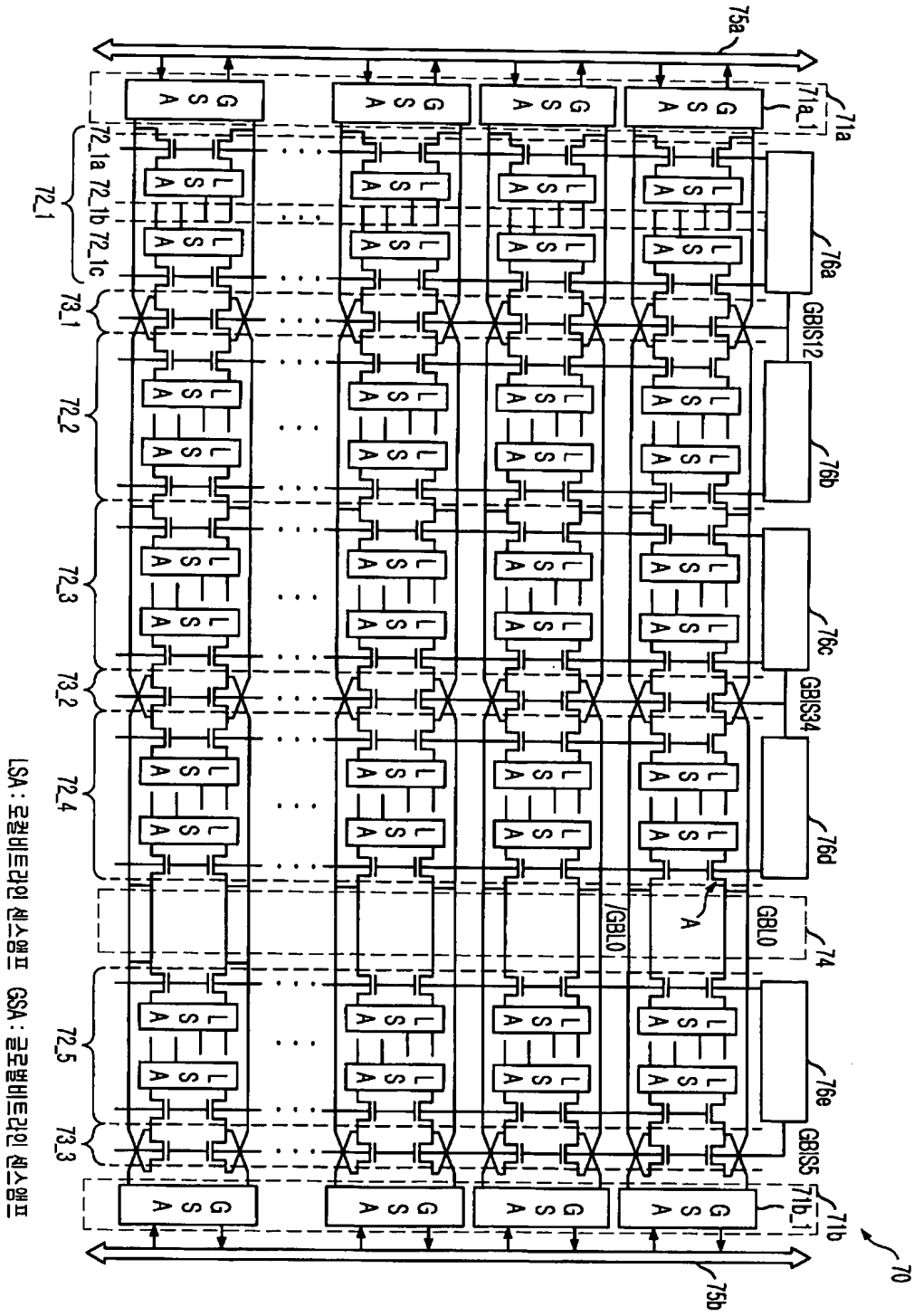


< 제1 셀블럭과 제2 셀블럭에 교대로 데이터 액세스가 일어날때 >

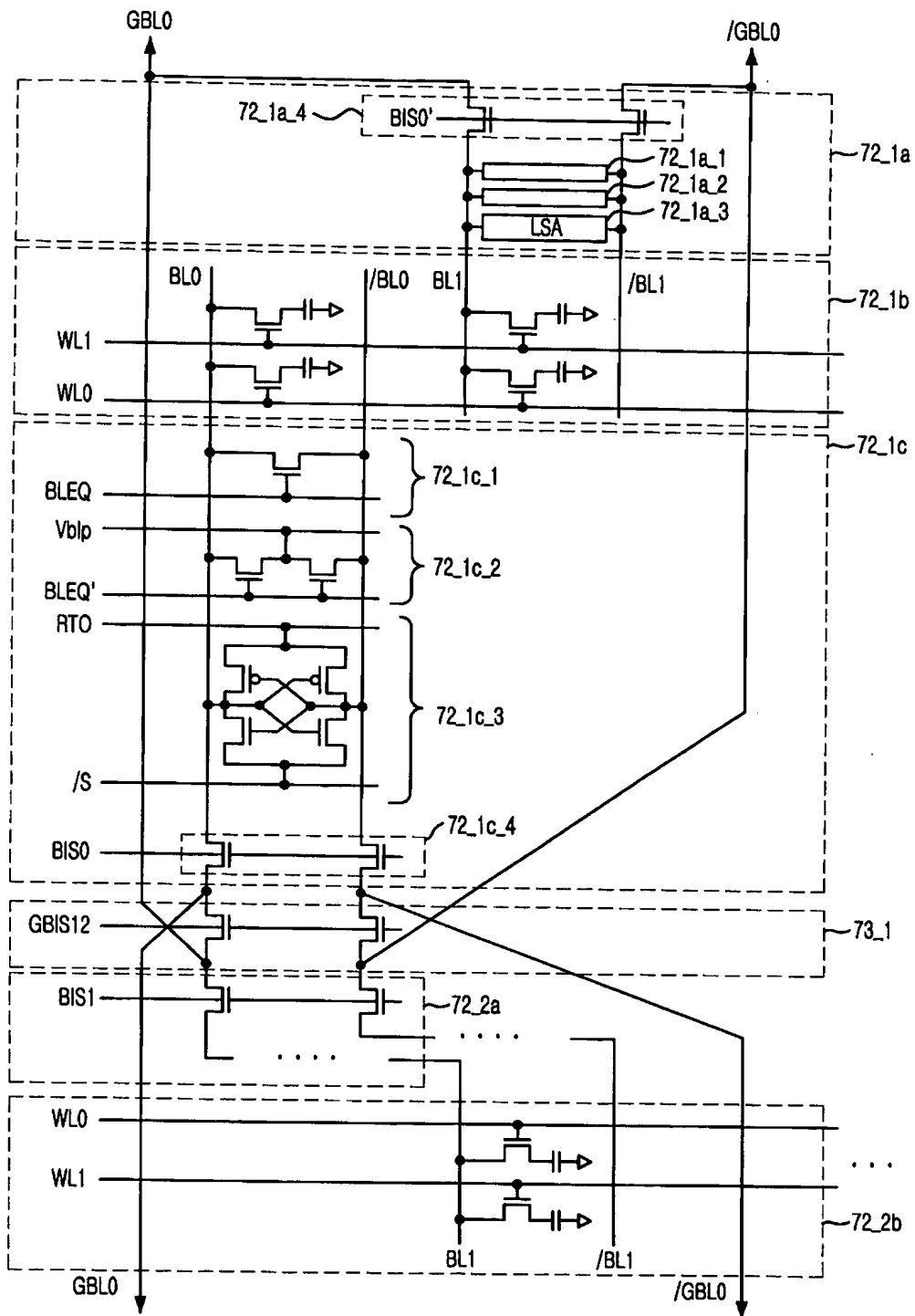
【도 8】



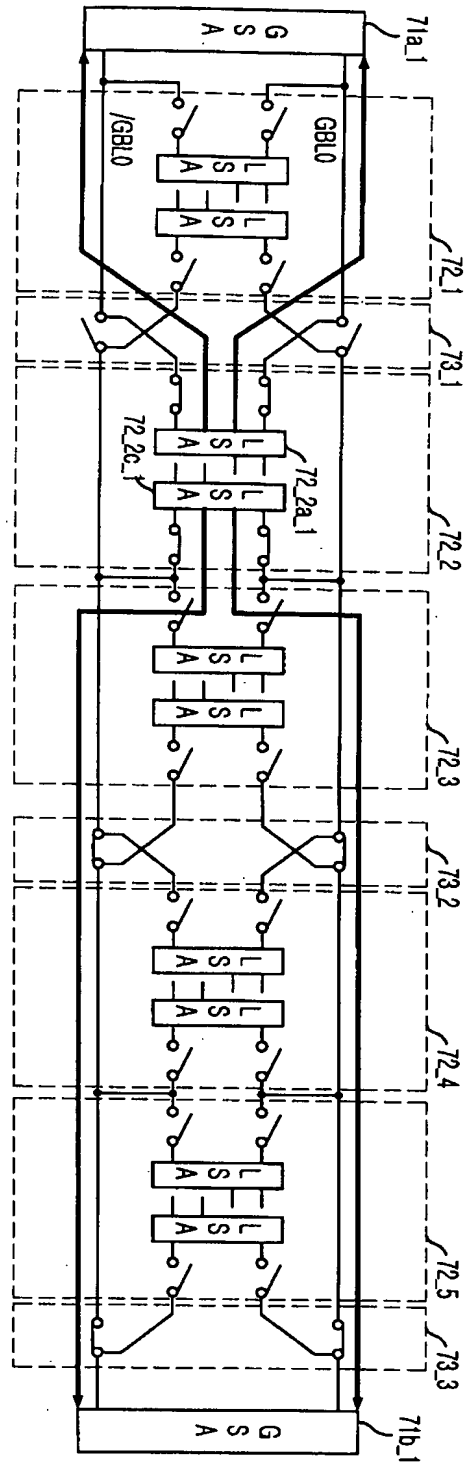
【도 9】



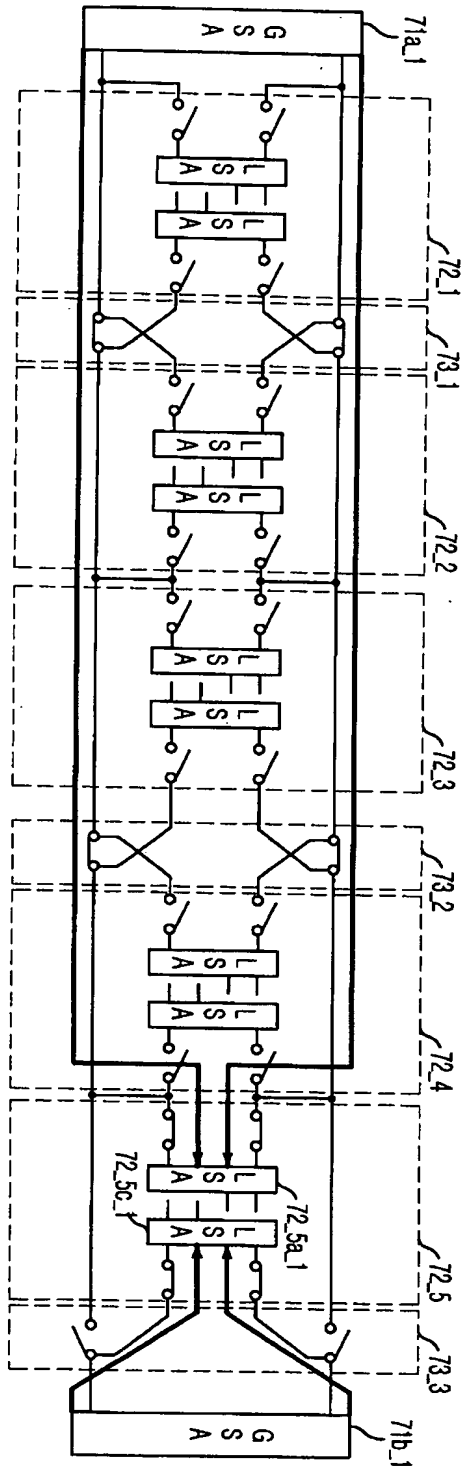
【도 10】



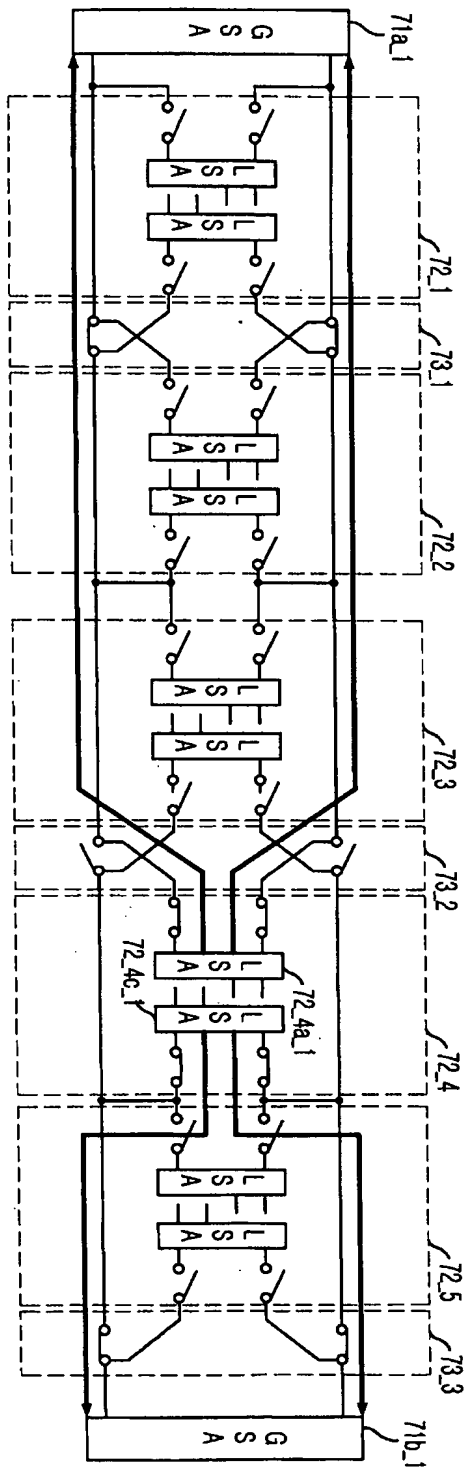
【도 11a】



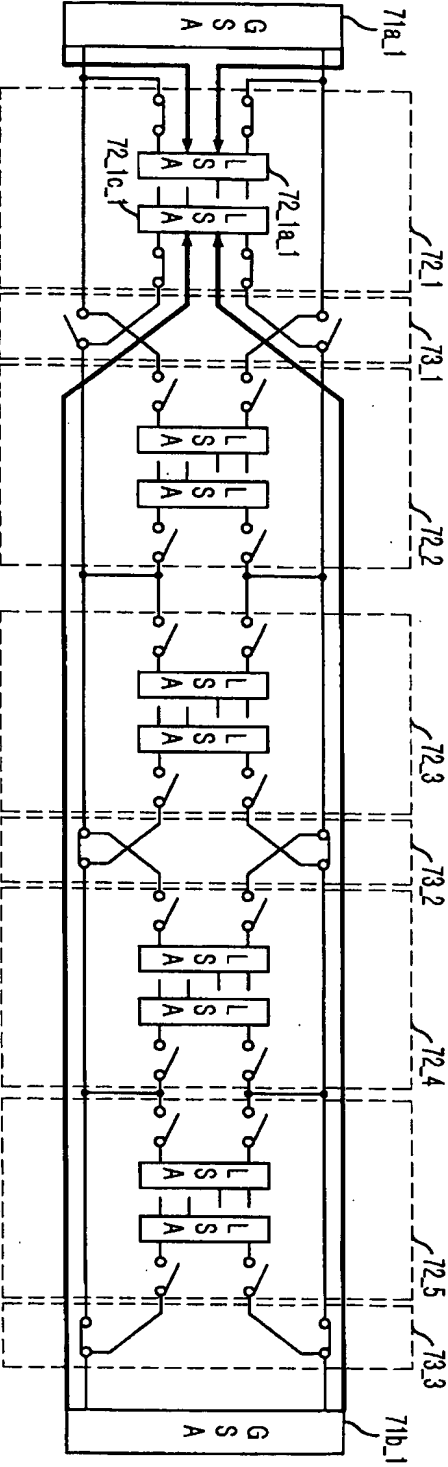
【도 11b】



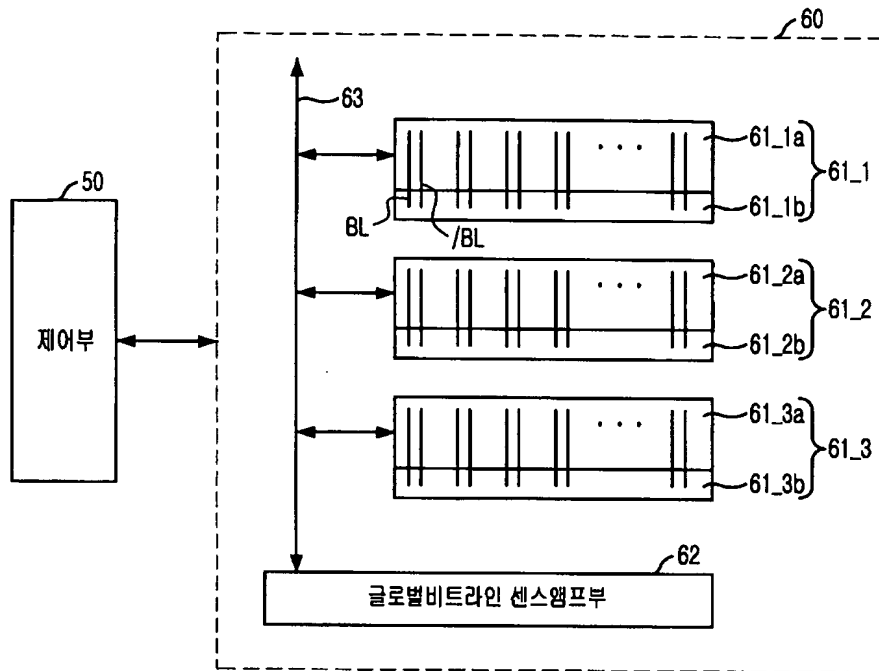
【도 11c】



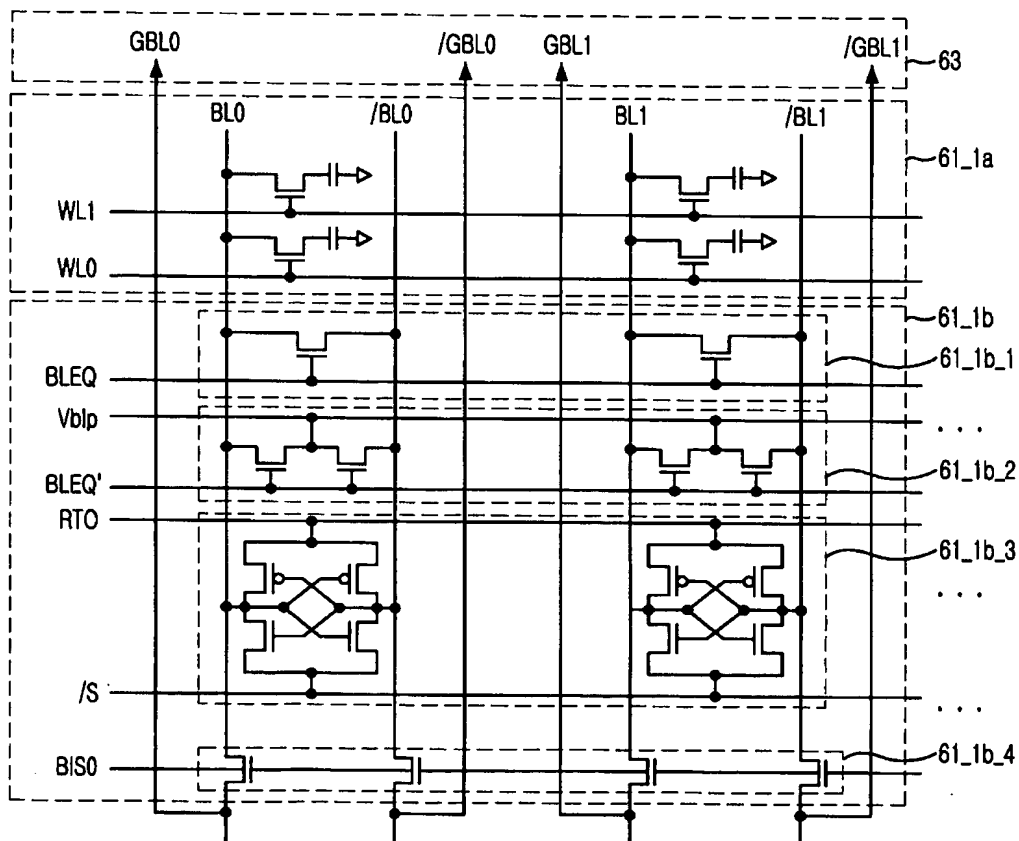
【도 11d】



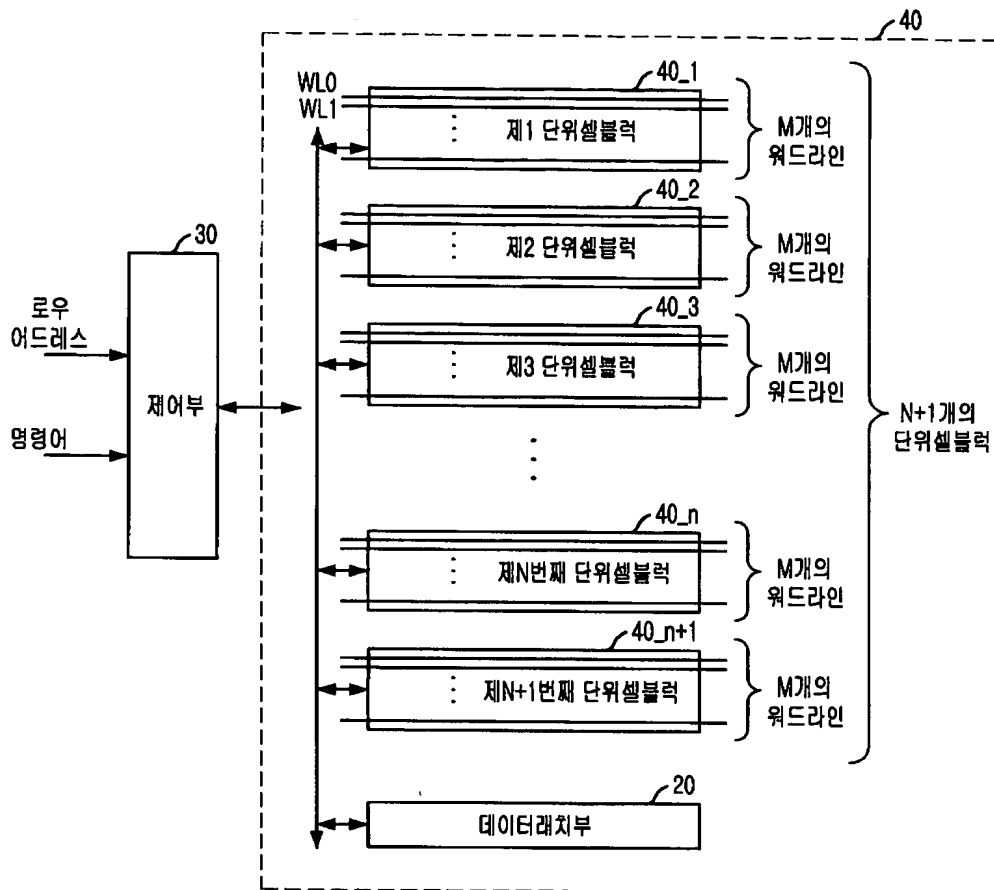
【도 12】



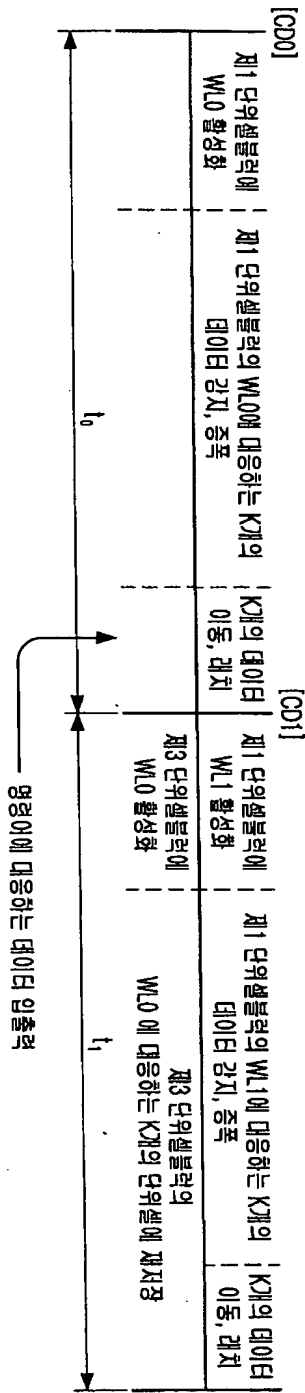
【도 13】



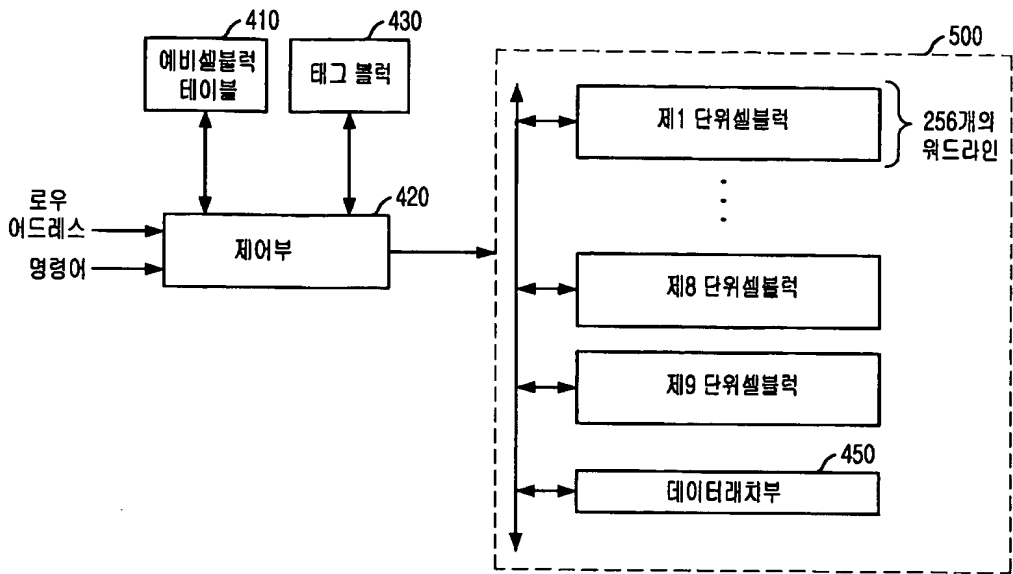
【도 14】



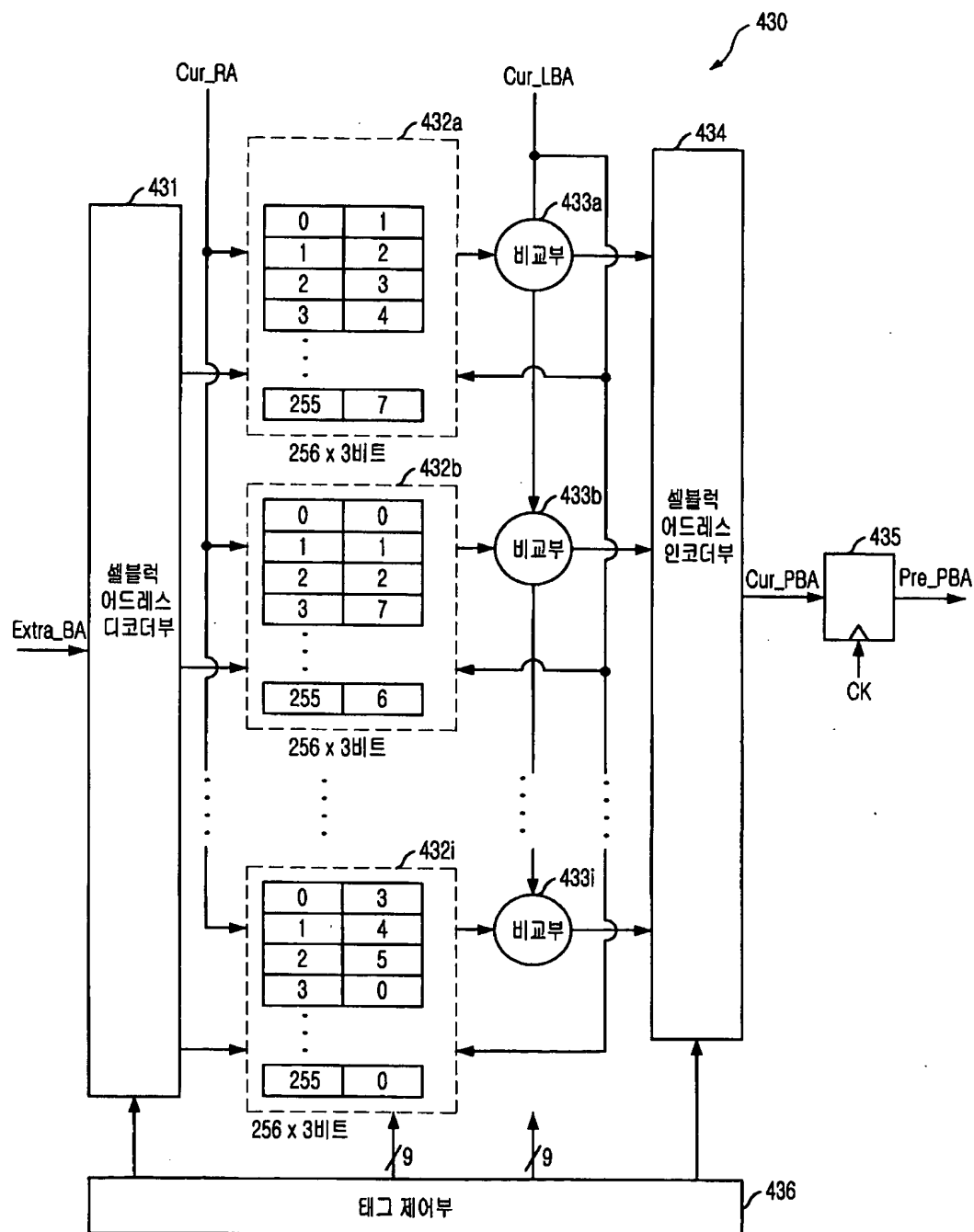
【도 15】



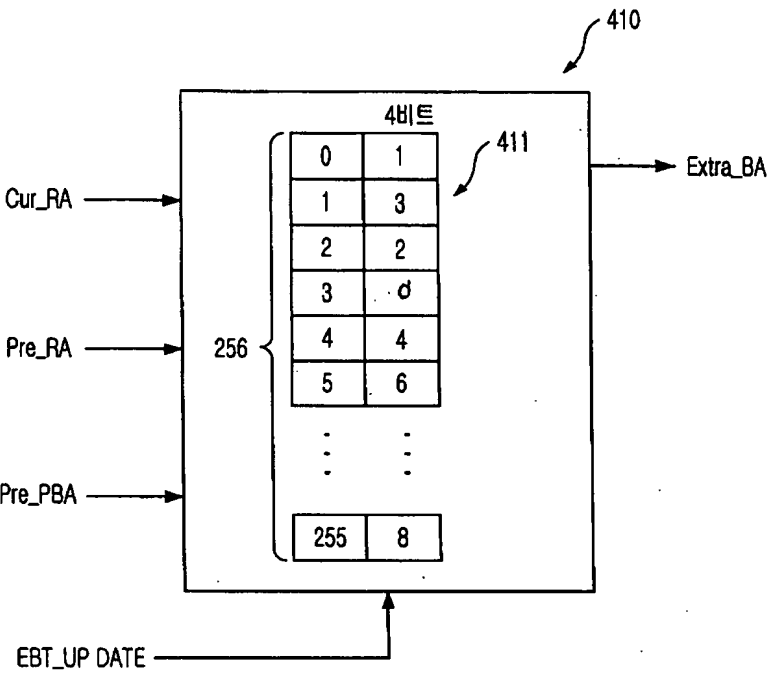
【도 16】



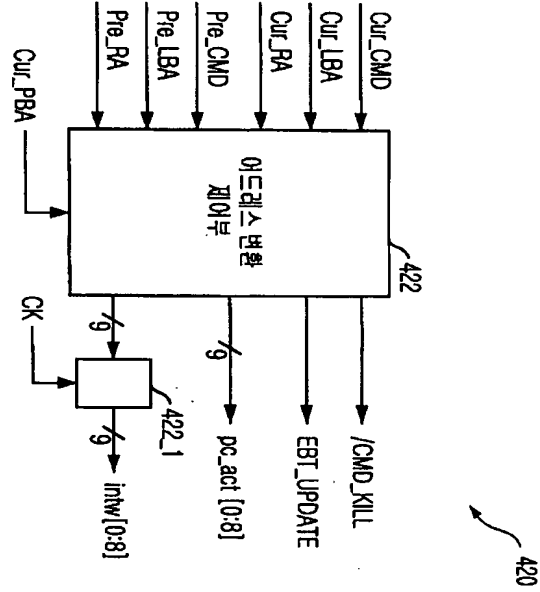
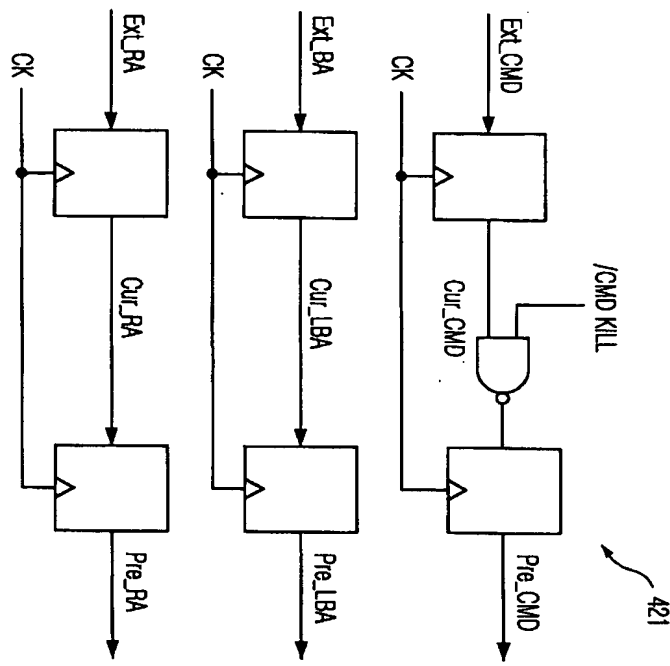
【도 17】



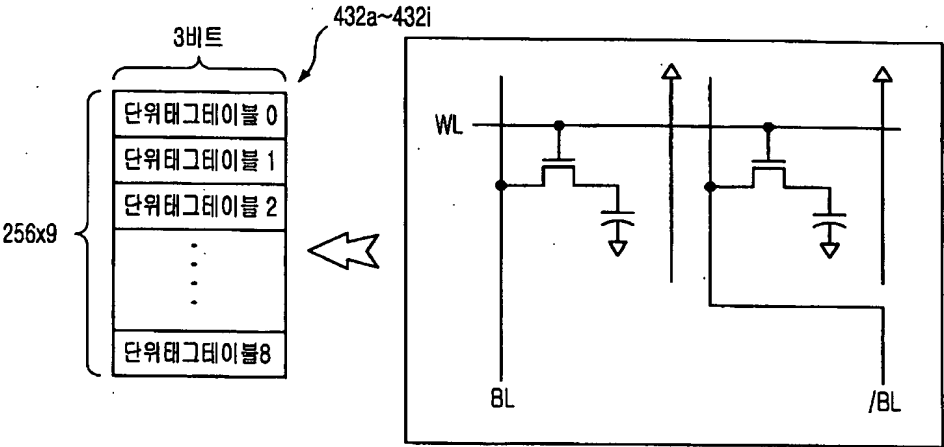
【도 18】



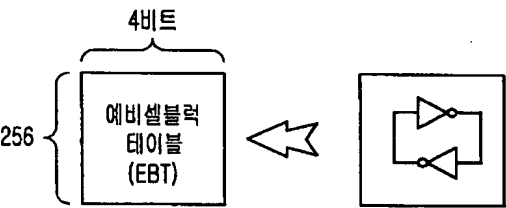
【도 19】



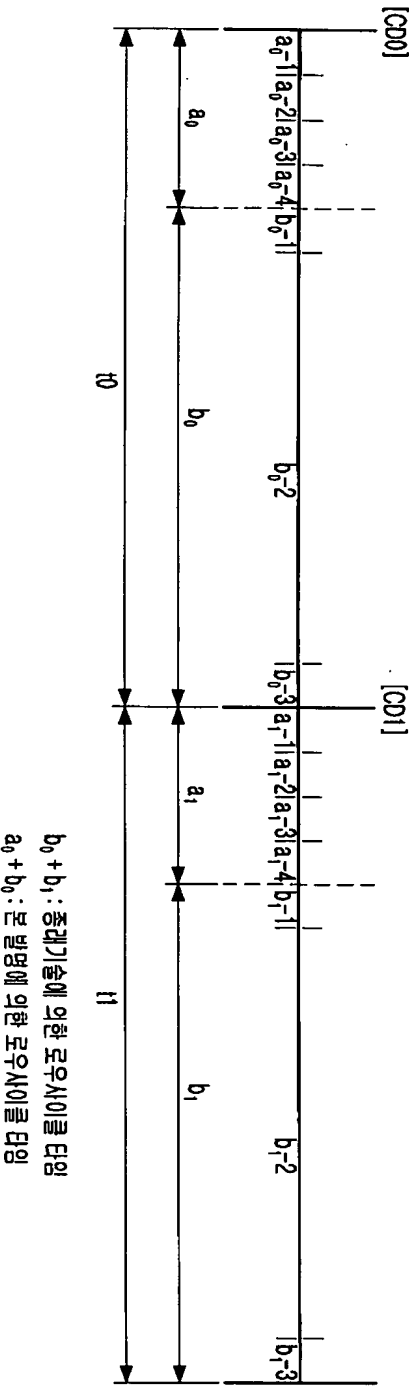
【도 20】



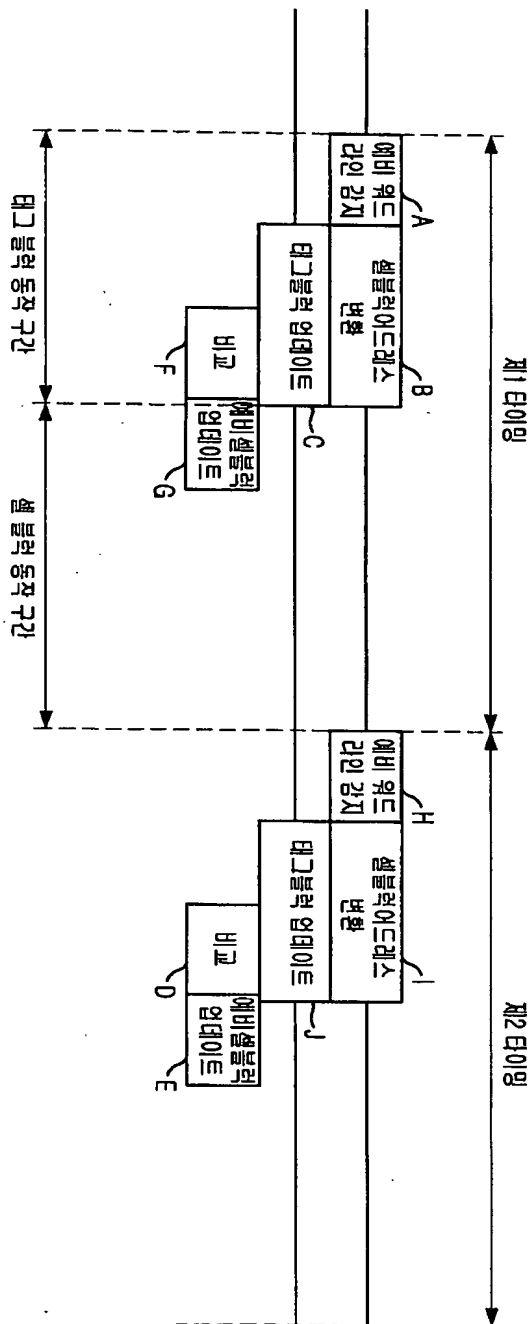
【도 21】



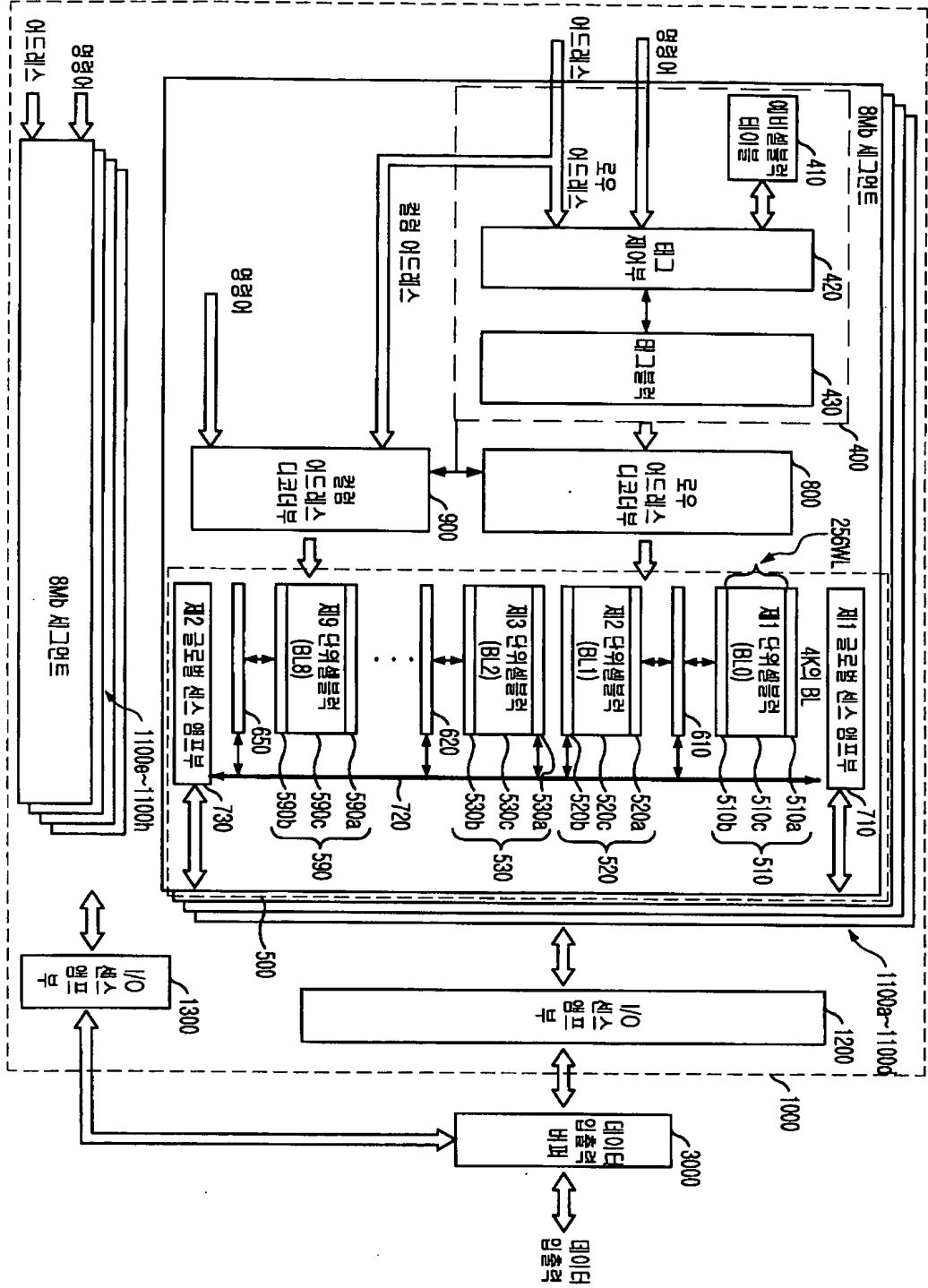
【도 22】



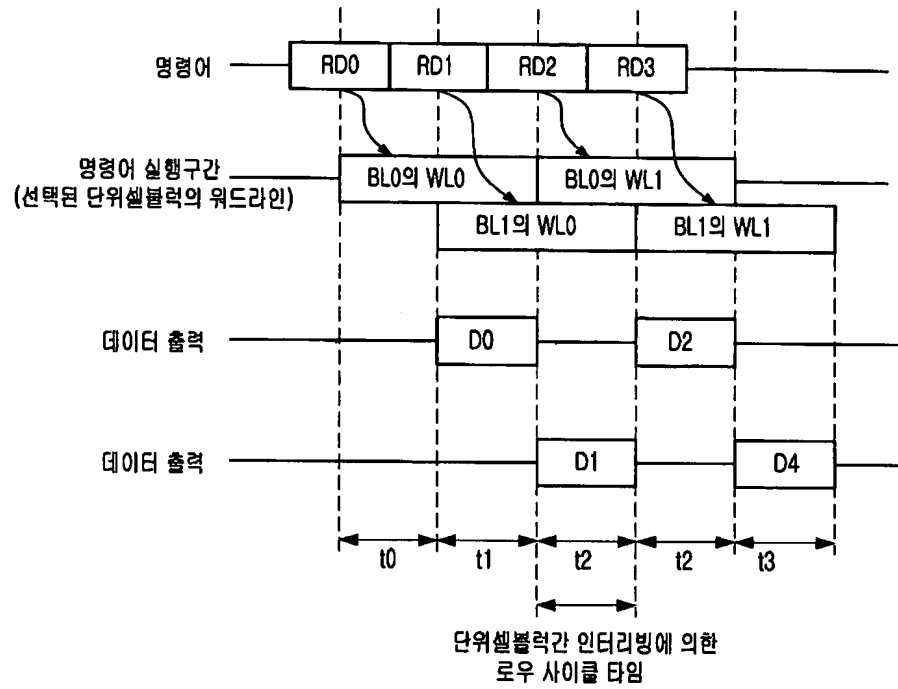
【도 23】



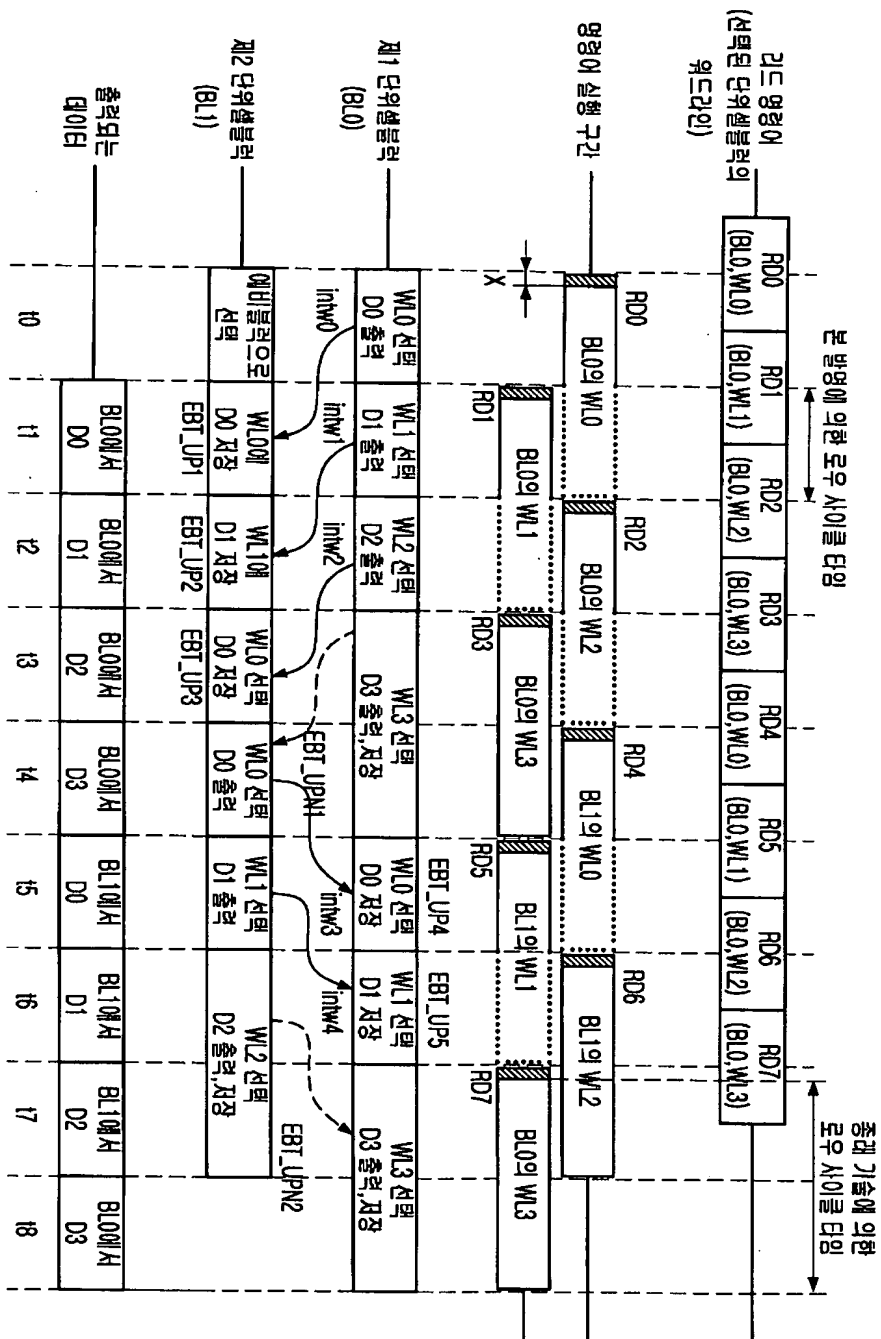
【도 24】



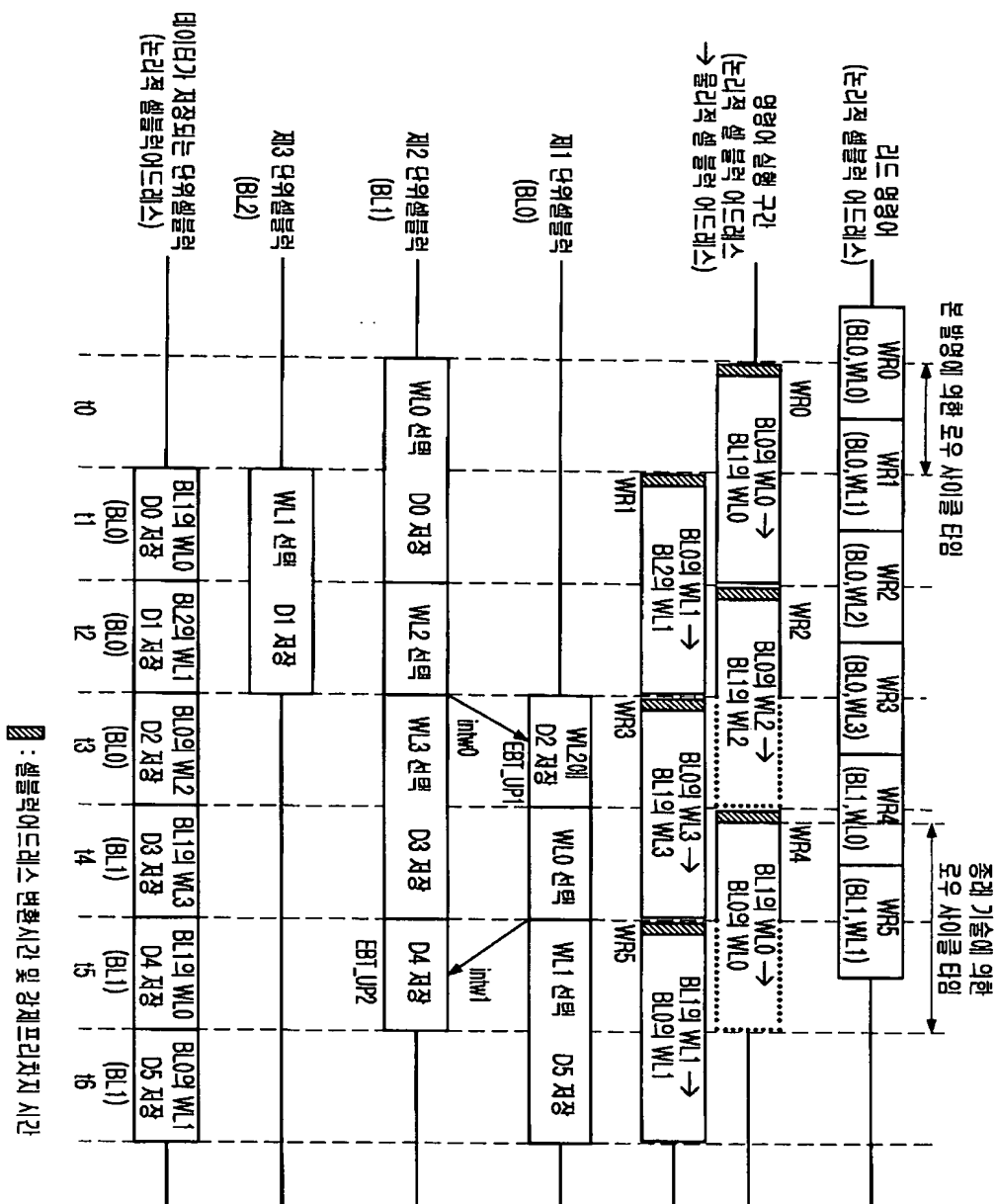
【도 25】



【도 26】

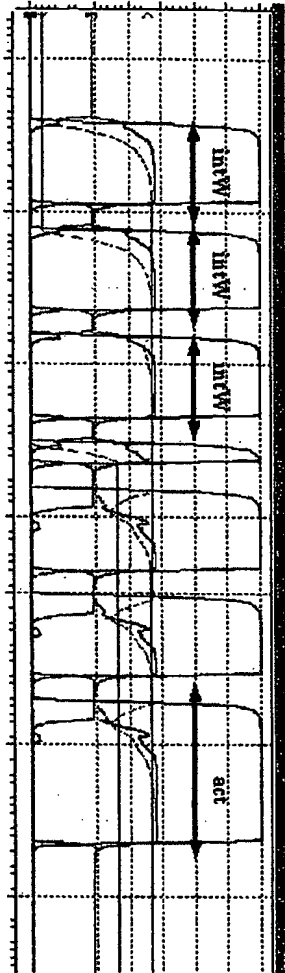


【도 27】

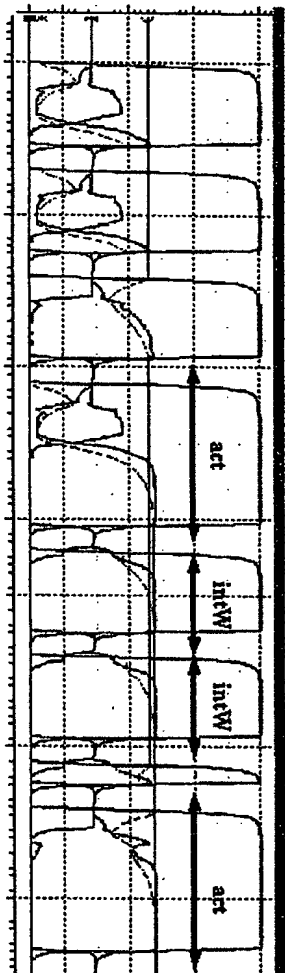


【도 28】

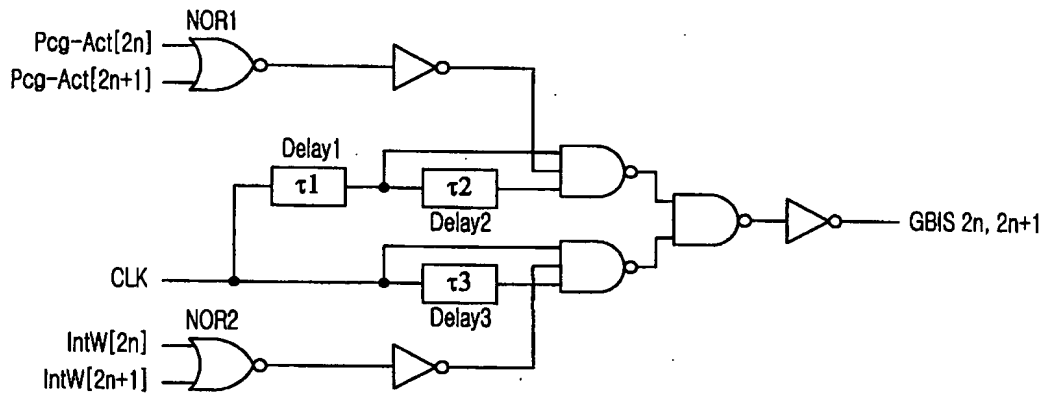
제2 단위 셀블럭



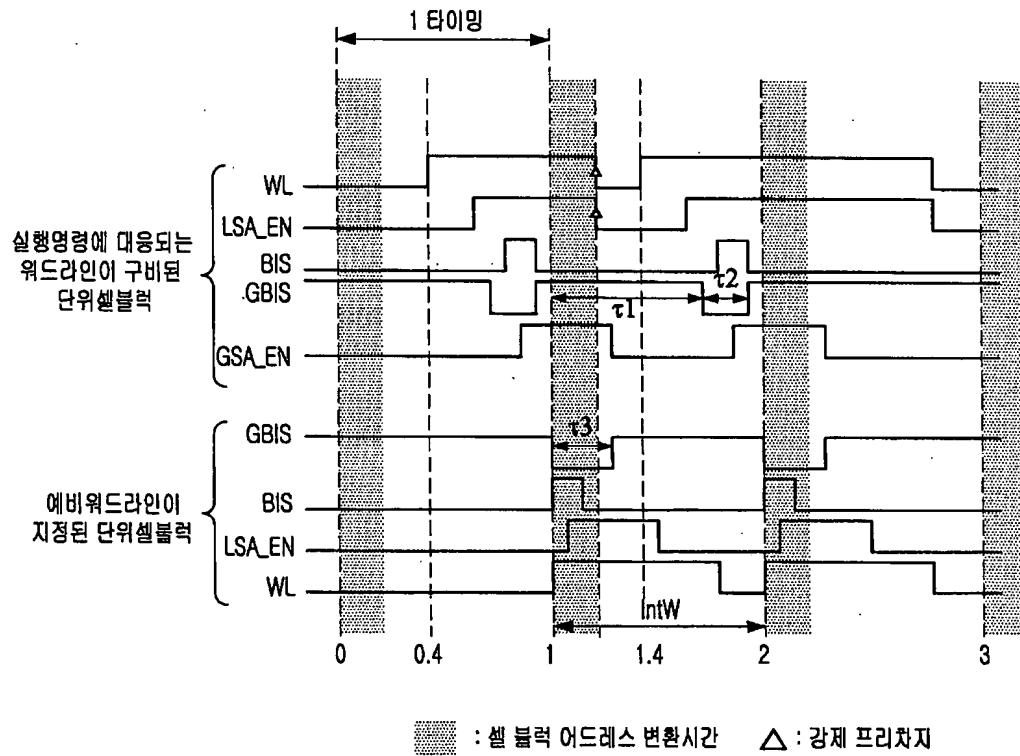
제1 단위 셀블럭



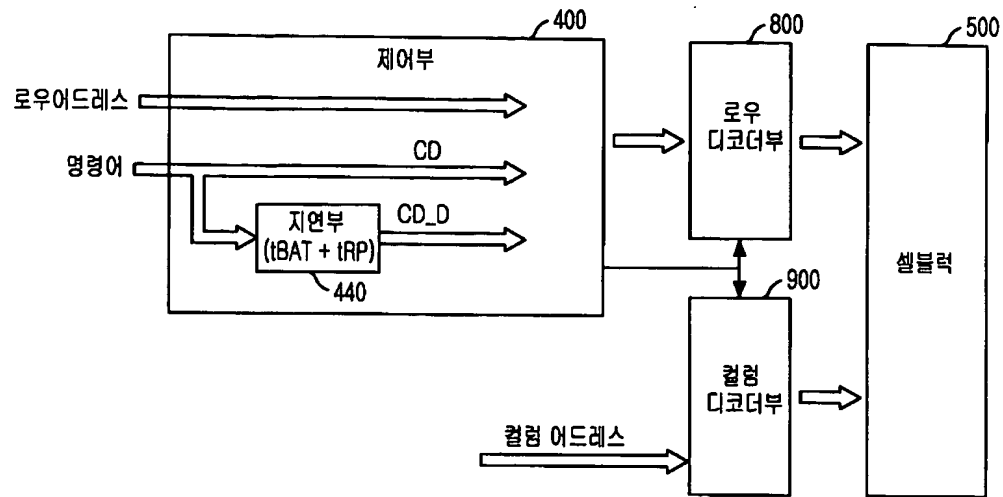
【도 29】



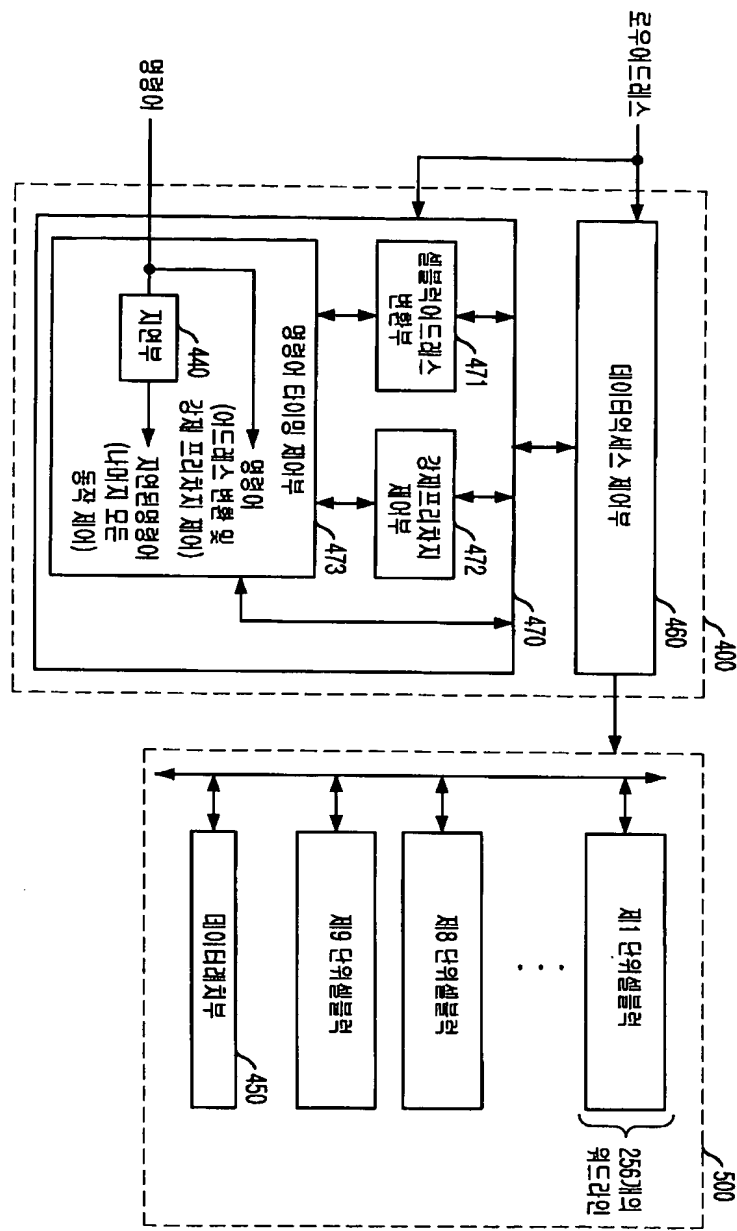
【도 30】



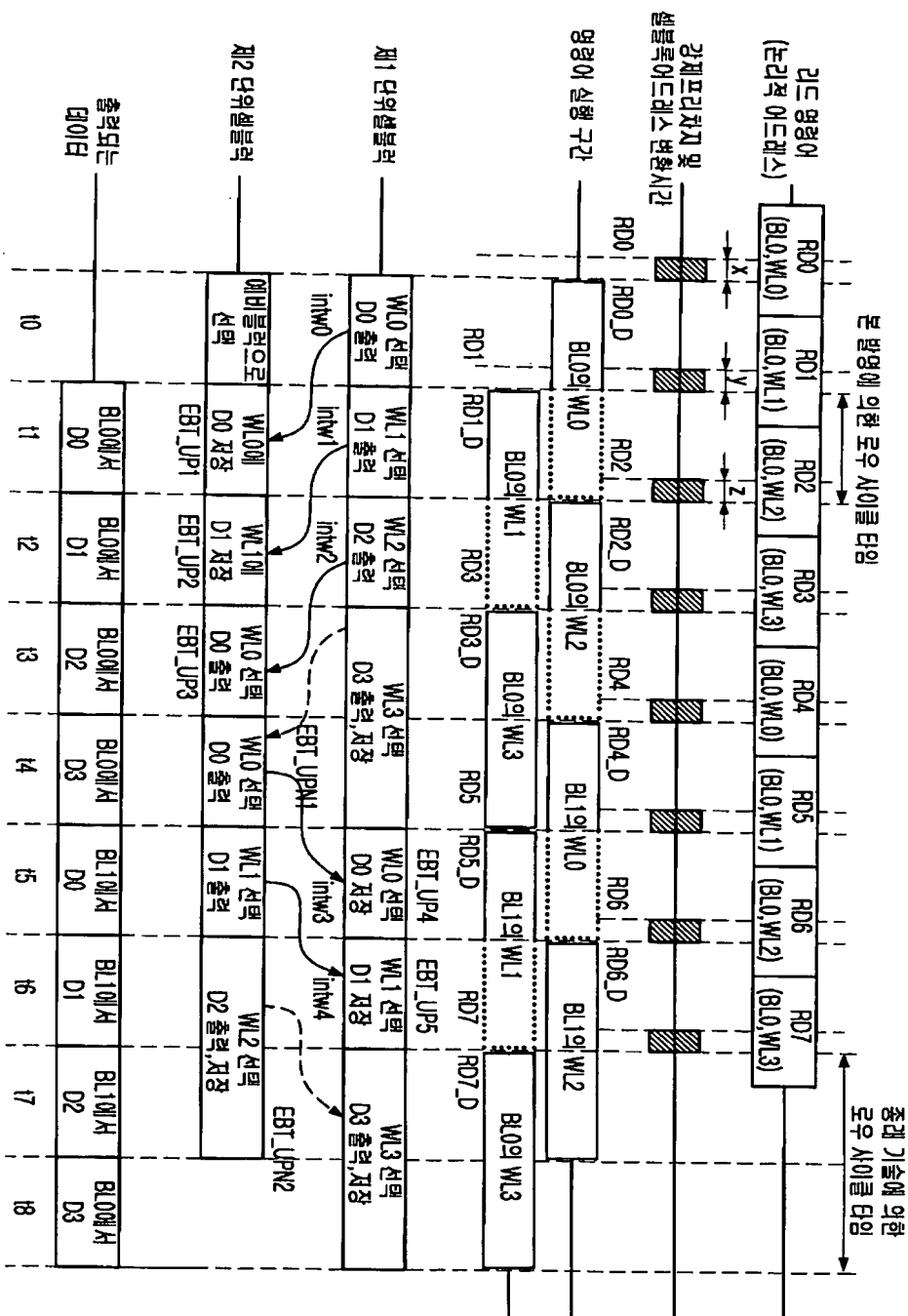
【도 31】



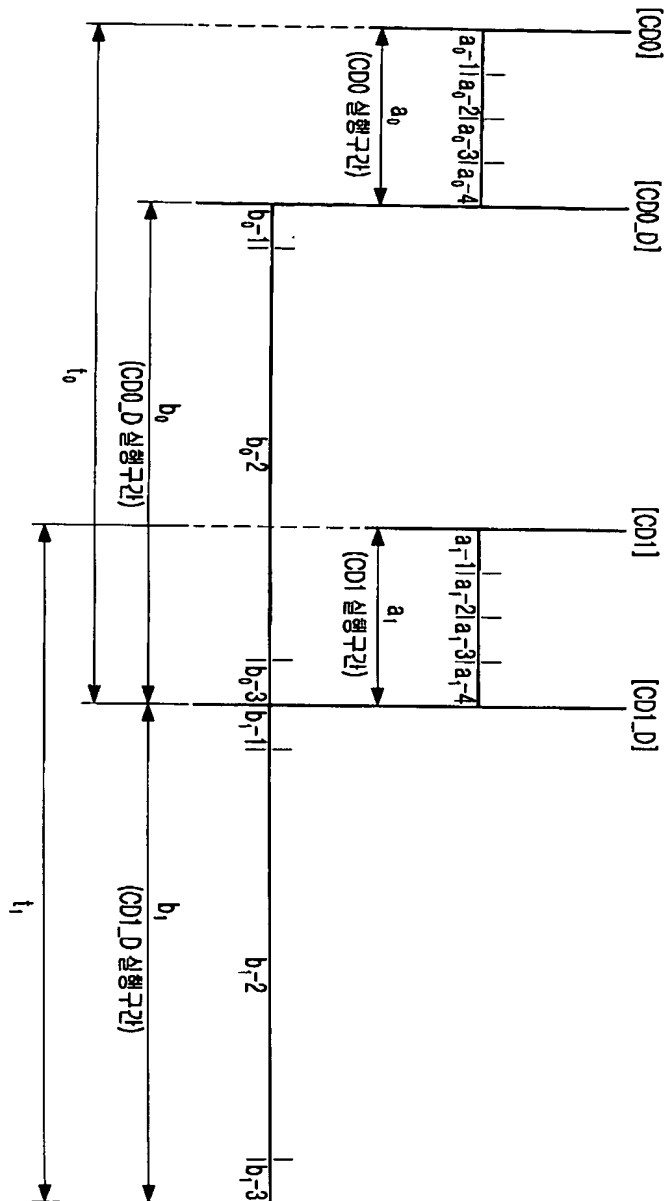
【도 32】



【도 33】



【도 34】



$b_0 + b_1$ (RC) : 종래기술에 의한 로우사이클 타입
 $b_0 \cdot 1/2$ (RC) : 본 발명에 의한 로우사이클 타입